

EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	DD	FFFFFFFFFFFFFFFF
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD		FFFFFFFFFFFFFFFF
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD		FFFFFFFFFFFFFFFF
EEE	DD	DD	FFF
EEE	DD	DD	FFF
EEE	DD	DD	FFF
EEE	DD	DD	FFF
EEE	DD	DD	FFF
EEE	DD	DD	FFF
EEEEEEEEEEEEEEEE	DD	DD	FFFFFFFFFFFFFFFF
EEEEEEEEEEEEEEEE	DD	DD	FFFFFFFFFFFFFFFF
EEEEEEEEEEEEEEEE	DD	DD	FFFFFFFFFFFFFFFF
EEE	DD	DD	FFF
EEE	DD	DD	FFF
EEE	DD	DD	FFF
EEE	DD	DD	FFF
EEE	DD	DD	FFF
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	DD	FFF
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD		FFF
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD		FFF

5
Va
--
00
00
00
00
00
00
00
00
00
7F
7F
7F
7F
7F
7F
7F
7F
7F

```

LL          IIIIII          SSSSSSSS
LL          IIIIII          SSSSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SSSSSS
LL          II             SSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```

```
0001      [ IDENT ('V04-000'),
0002      ( ++
0003      *****
0004      **
0005      **  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0006      **  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0007      **  ALL RIGHTS RESERVED.
0008      **
0009      **  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0010      **  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0011      **  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0012      **  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0013      **  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0014      **  TRANSFERRED.
0015      **
0016      **  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0017      **  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0018      **  CORPORATION.
0019      **
0020      **  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0021      **  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0022      **
0023      **
0024      *****
0025
0026
0027
0028
0029  FACILITY:      VAX/VMS EDF (EDIT/FDL) UTILITY
0030
0031  ABSTRACT:      This facility is used to create, modify, and optimize
0032                  FDL specification files.
0033
0034  ENVIRONMENT:   NATIVE/USER MODE
0035
0036  AUTHOR:        Ken F. Henderson Jr.
0037
0038  CREATION DATE: 27-Mar-1981
0039
0040  MODIFIED BY:
0041                V03-013 RRB0016      Rowland R. Bradley      6 Mar 1984
0042                  Signal error if insufficient information to do
0043                  analysis and disallow logging of file creation
0044                  if AUTO_TUNE (/NOINT)
0045
0046                V03-012 RRB0006      Rowland R. Bradley      12 Jan 1984
0047                  Enable user to specify analysis filename within optimize
0048                  script.
0049
0050                V03-011 KFH0011      Ken Henderson           8 Aug 1983
0051                  Changes for seperate compilation.
0052
0053                V03-010 KFH0010      Ken Henderson           26 Apr 1983
0054                  Modified SET_PROC to set VISIBLE_QUESTION.
0055                  REDESIGN => TOUCHUP.
0056
0057                V03-009 KFH0009      Ken Henderson           14 Apr 1983
```

EDFFUNCS
V04-000

Source Listing

K 14
16-Sep-1984 01:17:14
5-Sep-1984 13:37:08

VAX-11 Pascal V2.4-277
DISK\$VMSMASTER:[EDF.SRC]EDFFUNCS.PAS;1 (1) Page 2

0058
0059
0060
0061
0062
0063
0064
0065
0066
0067
0068
0069
0070
0071
0072
0073
0074
0075
0076
0077
0078
0079
0080
0081
0082
0083
0084
0085
0086
0087
0088
0089
0090
0091

Added SET PROC.
Removed DESIGN_SCRIPT_PROC.

V03-008 KFH0008 Ken Henderson 20 Jan 1983
Removed references to DASH.

V03-007 KFH0007 Ken Henderson 11 Jan 1983
Modified CREATE_NEW_FDL to output
"Output not created" message on one
line, in reverse video, with bell

V03-006 KFH0006 Ken Henderson 15 Nov 1982
Added support for Pascal V2

V03-005 KFH0005 Ken Henderson 8 Sept 1982
Modified call to Script_option to
use new QUERY routine.

V03-004 KFH0004 Ken Henderson 31 March 1982
Modified CREATE_NEW_FDL to fix
FT2 QAR 967

V03-003 KFH0003 Ken Henderson 28 March 1982
Modified CREATE_NEW_FDL to not output
FDL file if the definition is empty.

V03-002 KFH0002 Ken Henderson 23-Mar-1982
Modified HELP_PROC to fix FT2 QAR 831

V03-001 KFH0001 Ken Henderson 17-Mar-1982
Modified a few routines to fix FT2
QARs 500,510

-- }

EDFFUNCS
V04-000

Source Listing

L 14
16-Sep-1984 01:17:14
5-Sep-1984 13:37:08

VAX-11 Pascal V2.4-277
DISK\$VMSMASTER:[EDF.SRC]EDFFUNCS.PAS;1 (2) Page 3

```
0093 ENVIRONMENT ('LIB$:EDFFUNCS'),
0094
0095 INHERIT (
0096
0097   'SYSS$LIBRARY:STARLET',
0098   'SHR$LIBS:FDPARDEF',
0099   'LIB$:EDFSDLMSG',
0100   'LIB$:EDFSTRUCT',
0101   'LIB$:EDFCONST',
0102   'LIB$:EDFTYPE',
0103   'LIB$:EDFVAR',
0104   'LIB$:EDFEXTERN',
0105   'LIB$:EDFCHF',
0106   'LIB$:EDFUTIL',
0107   'LIB$:EDFASK',
0108   'LIB$:EDFSHOW',
0109   'LIB$:EDFDESIGN'
0110 )
0111
0112 MODULE EDFFUNCS (INPUT,OUTPUT);
0113
```

EDFFUNCS
V04-000

Source Listing

M 14
16-Sep-1984 01:17:14
5-Sep-1984 13:37:08

VAX-11 Pascal V2.4-277
DISK\$VMSMASTER:[EDF.SRC]EDFFUNCS.PAS;1 (3) Page 4

```
0115 { ++
0116
0117 CREATE_NEW_FDL -- Routine to output a new FDL file.
0118
0119 This routine outputs the FDL file to the disk.
0120
0121 CALLING SEQUENCE:
0122
0123 CREATE_NEW_FDL;
0124
0125 INPUT PARAMETERS:
0126
0127 none
0128
0129 IMPLICIT INPUTS:
0130
0131 none
0132
0133 OUTPUT PARAMETERS:
0134
0135 none
0136
0137 IMPLICIT OUTPUTS:
0138
0139 none
0140
0141 ROUTINES CALLED:
0142
0143 none
0144
0145 ROUTINE VALUE:
0146
0147 none
0148
0149 SIGNALS:
0150
0151 none
0152
0153 SIDE EFFECTS:
0154
0155 none
0156 -- }
0157
```

```
0159  PROCEDURE CREATE_NEW_FDL;
0160
0161  VAR
0162      TEMP_STRING255      : STRING255;
0163      FID_BLOCK           : ARRAY [0..2] OF LONG;
0164      I                   : INTEGER;
0165      J                   : INTEGER;
0166
0167  BEGIN
0168
0169      { +
0170      Only output the FDL file if the definition is not empty.
0171      - }
0172      IF DEF_HEAD = DEF_TAIL THEN
0173
0174          BEGIN
0175
0176              FILE_CREATED      := FALSE;
0177
0178              WRITELN (CRLF,SHIFT,CONTROL_G,ANSI_REVERSE,
0179              'Output not created - Current FDL Definition empty.',ANSI_RESET);
0180
0181              END { IF TRUE DEF_HEAD = DEF_TAIL }
0182
0183          ELSE
0184
0185              BEGIN
0186
0187                  RES_OUTPUT_FILENAME_DESC := NULL_STRING;
0188                  NEW (RES_OUTPUT_FILENAME_DESC.DSC$A_POINTER);
0189                  RES_OUTPUT_FILENAME_DESC.DSC$W_LENGTH := 255;
0190                  FLAGS.FDL$V_SIGNAL := TRUE;
0191                  FLAGS.FDL$V_CALLBACK := FALSE;
0192
0193                  ISTATUS := FDL$CREATE (
0194
0195                      NL_DEV_DESC,
0196                      OUTPUT_FILENAME_DESC,
0197                      DEFAULT_FILENAME_DESC,
0198                      RES_OUTPUT_FILENAME_DESC,
0199                      FID_BLOCK,
0200                      FLAGS
0201                  );
0202
0203                  IF ODD (ISTATUS) THEN
0204
0205                      BEGIN
0206
0207                          { +
0208                          Open his file and initialize it.
0209                          - }
0210                          DEST_IS_TERMINAL := FALSE;
0211
0212                          WITH RES_OUTPUT_FILENAME_DESC DO
0213
0214                              BEGIN
0215
0216                                  FOR I := 1 TO 255 DO
```

```
0216
0217         IF I > DSC$W_LENGTH THEN
0218             TEMP_STRING255[I]      := ' '
0219
0220         ELSE
0221             TEMP_STRING255[I]      := DSC$A_POINTER^[I];
0222
0223     END;
0224
0225     { +
0226     Clear out the terminal in case the terminal is the output.
0227     - }
0228     IF NOT AUTO_TUNE THEN
0229
0230     BEGIN
0231
0232         OPEN (FDL_DEST,SYSS$OUTPUT_NAME,NEW,RECORD_LENGTH := 252);
0233         CLOSE (FDL_DEST);
0234
0235     END;
0236
0237     { +
0238     Now implement 'granularity'.
0239     - }
0240     IF ISAM_ORG THEN
0241
0242         SHUFFLE_AREAS;
0243
0244     { +
0245     Now open the 'real' FDL file.
0246     - }
0247     OPEN (FDL_DEST,TEMP_STRING255,OLD);
0248     REWRITE (FDL_DEST);
0249
0250     { +
0251     Put the current definition out to the disk.
0252     - }
0253     GENERATE_FDL;
0254
0255     { +
0256     We're done, close the file.
0257     - }
0258     CLOSE (FDL_DEST);
0259
0260     { +
0261     Setup to show the created filename on exit.
0262     - }
0263     FILE_CREATED      := TRUE;
0264
0265     ( IF AUTO_TUNE THEN
0266         EDF$RESET_SCROLL;
0267     )
0268
0269     END; { IF ODD (ISTATUS) }
```

EDFFUNCS
V04-000

Source Listing

C 15
16-Sep-1984 01:17:14
5-Sep-1984 13:37:08

VAX-11 Pascal V2.4-277
DISK\$VMSMASTER:[EDF.SRC]EDFFUNCS.PAS;1 (4) Page 7

0273 END; { IF FALSE DEF_HEAD = DEF_TAIL }
0274
0275 END; { CREATE_NEW_FDL }

```
0277 { ++
0278
0279 ADD_FDL_LINE -- Let user add a line_object to the Definition Linked List.
0280
0281 This routine prompts the user for his information and puts it into the
0282 Deifinition.
0283
0284 CALLING SEQUENCE:
0285
0286 ADD_FDL_LINE;
0287
0288 INPUT PARAMETERS:
0289
0290 none
0291
0292 IMPLICIT INPUTS:
0293
0294 SYSS$INPUT:
0295
0296 OUTPUT PARAMETERS:
0297
0298 none
0299
0300 IMPLICIT OUTPUTS:
0301
0302 The Definition Linked List
0303 DEF_CURRENT
0304
0305 ROUTINES CALLED:
0306
0307 none
0308
0309 ROUTINE VALUE:
0310
0311 none
0312
0313 SIGNALS:
0314
0315 none
0316
0317 SIDE EFFECTS:
0318
0319 none
0320
0321 -- }
```

```
0323  PROCEDURE ADD_FDL_LINE;
0324
0325  VAR
0326      DEF_TEST      : ^LINE_OBJECT;
0327      SAVE_CURRENT  : ^LINE_OBJECT;
0328      SAVE          : LINE_OBJECT;
0329      FOUND_PRI     : BOOLEAN;
0330      EXISTS        : BOOLEAN;
0331      PROCEED       : BOOLEAN;
0332
0333  BEGIN
0334
0335      SAVE.STRING     := NULL_STRING;
0336      TEST.STRING     := NULL_STRING;
0337
0338      FULL_CHOICE     := TRUE;
0339      QUERY (EDF$K_TEST_PRIMARY);
0340
0341      FULL_CHOICE     := TRUE;
0342      ASK_TEST_SECONDARY;
0343
0344      SAVE            := TEST;
0345
0346      { +
0347      Setup to display definition on the terminal.
0348      - }
0349      OPEN            (FDL_DEST,SY$OUTPUT_NAME,NEW,RECORD_LENGTH := 252);
0350      REWRITE         (FDL_DEST);
0351
0352      NEW (DEF_TEST);
0353      DEF_TEST*       := TEST;
0354      DEF_TEST^.FORE  := NIL;
0355      DEF_TEST^.BACK  := NIL;
0356      SAVE_CURRENT    := DEF_CURRENT;
0357      DEF_CURRENT     := DEF_TEST;
0358
0359      SHOW_CUR_PRI_SEC (FALSE);
0360
0361      DEF_CURRENT     := SAVE_CURRENT;
0362      DISPOSE (DEF_TEST);
0363
0364      CLOSE          (FDL_DEST);
0365
0366      EXISTS         := FIND_OBJECT (
0367      SAVE.OBJECT_TYPE,SAVE.PRIMARY,SAVE.PRINUM,SAVE.SECONDARY,SAVE.SECNUM);
0368
0369      IF EXISTS THEN
0370
0371          PROCEED     := QUERY (EDF$K_CONFIRM)
0372
0373      ELSE
0374
0375          PROCEED     := TRUE;
0376
0377      IF PROCEED THEN
0378
0379          BEGIN
```

```
0380
0381     TEST      := SAVE;
0382
0383     ASK_TEST_SECONDARY_VALUE;
0384
0385     MAKE_SCRATCH;
0386
0387     DEF_SCRATCH^ := TEST;
0388
0389     IF DEF_SCRATCH^.PRIMARY = TITLE THEN
0390
0391         DEF_SCRATCH^.OBJECT_TYPE      := PRI
0392
0393     ELSE
0394
0395         DEF_SCRATCH^.OBJECT_TYPE      := SEC;
0396
0397 { **** SUPPORT END OF LINE COMMENTS !!! *** }
0398
0399     INSERT_IN_ORDER (REPLACE_OBJ);
0400
0401     IF TEST.PRIMARY <> TITLE THEN
0402
0403     BEGIN
0404
0405         { +
0406         If there wasn't one of these primaries, make one.
0407         - }
0408         DEF_CURRENT := DEF_HEAD;
0409         FOUND_PRI   := FALSE;
0410
0411         REPEAT
0412
0413             IF (
0414                 (DEF_CURRENT^.OBJECT_TYPE = PRI)
0415                 AND
0416                 (DEF_CURRENT^.PRIMARY = SAVE.PRIMARY)
0417                 AND
0418                 (DEF_CURRENT^.PRINUM = SAVE.PRINUM)
0419             ) THEN
0420
0421                 FOUND_PRI   := TRUE
0422
0423             ELSE
0424
0425                 INCR_CURRENT;
0426
0427         UNTIL (FOUND_PRI OR (DEF_CURRENT = NIL));
0428
0429         IF NOT FOUND_PRI THEN
0430
0431         BEGIN
0432
0433             TEST.OBJECT_TYPE      := PRI;
0434             TEST.PRIMARY          := SAVE.PRIMARY;
0435             TEST.PRINUM           := SAVE.PRINUM;
0436             TEST.SECONDARY        := DUMMY_SECONDARY$;
```

EDFFUNCS
V04-000

Source Listing

G 15
16-Sep-1984 01:17:14
5-Sep-1984 13:37:08

VAX-11 Pascal V2.4-277
DISK\$VMSMASTER:[EDF.SRC]EDFFUNCS.PAS;1 (6) Page 11

```
0437      TEST.SECNUM      := 0;
0438      TEST.STRING      := NULL_STRING;
0439      TEST.COMMENT     := NULL_STRING;
0440
0441      MAKE_SCRATCH;
0442
0443      DEF_SCRATCH^      := TEST;
0444
0445      INSERT_IN_ORDER (REPLACE_OBJ);
0446
0447      END;      { IF NOT FIND_OBJECT }
0448
0449      END;      { IF TEST.PRIMARY <> TITLE }
0450
0451      CLEAR (SCREEN);
0452
0453      WRITELN (SHIFT,TAB,TAB,ANSI_REVERSE,
0454      ' Resulting Primary Section ',
0455      ANSI_RESET,CRLF);
0456
0457      OPEN      (FDL_DEST,SYSS$OUTPUT_NAME,NEW,RECORD_LENGTH := 252);
0458      REWRITE (FDL_DEST);
0459
0460      SHOW_PRIMARY_SECTION (SAVE);
0461
0462      CLOSE (FDL_DEST);
0463
0464      END;      { IF TRUE PROCEED }
0465
0466      TEST      := SAVE;
0467
0468      QUERY (EDF$K_RETURN);
0469
0470      END;      { ADD_FDL_LINE }
```

```
0472      ( ++
0473
0474      CHECK_DEFAULT -- See if the current default primary exists.
0475
0476      This routine searches the definition and checks to make sure that
0477      the current default is OK.
0478
0479      CALLING SEQUENCE:
0480
0481      CHECK_DEFAULT;
0482
0483      INPUT PARAMETERS:
0484
0485      none
0486
0487      IMPLICIT INPUTS:
0488
0489
0490      OUTPUT PARAMETERS:
0491
0492      none
0493
0494      IMPLICIT OUTPUTS:
0495
0496
0497      ROUTINES CALLED:
0498
0499
0500      ROUTINE VALUE:
0501
0502      none
0503
0504      SIGNALS:
0505
0506      none
0507
0508      SIDE EFFECTS:
0509
0510      none
0511
0512      -- }
```

```
0514  PROCEDURE CHECK_DEFAULT;
0515
0516  VAR
0517      FOUND_PRIMARY      : BOOLEAN;
0518
0519  BEGIN
0520
0521      IF DEF_HEAD <> DEF_TAIL THEN
0522      BEGIN
0523
0524          { +
0525          Does the current default primary exist?
0526          - }
0527          DEF_CURRENT      := DEF_HEAD;
0528          TEST.OBJECT_TYPE := PRI;
0529          TEST.PRIMARY     := DEFAULT_PRIMARY;
0530          TEST.PRINUM      := DEFAULT_PRINUM;
0531          FOUND_PRIMARY    := FALSE;
0532
0533          REPEAT
0534
0535              IF CURRENT_EQ_TEST (TEST,FALSE) THEN
0536                  FOUND_PRIMARY := TRUE
0537
0538              ELSE
0539                  INCR_CURRENT;
0540
0541          UNTIL (DEF_CURRENT = NIL) OR FOUND_PRIMARY;
0542
0543          IF NOT FOUND_PRIMARY THEN
0544          BEGIN
0545
0546              { +
0547              Find out what the 1st 'real' primary is.
0548              - }
0549              DEF_CURRENT      := DEF_HEAD;
0550
0551              IF DEF_CURRENT^.PRIMARY = IDENT THEN
0552                  INCR_CURRENT;
0553
0554              { +
0555              Set the default up to be the first one that exists.
0556              - }
0557              DEFAULT_PRIMARY := DEF_CURRENT^.PRIMARY;
0558              DEFAULT_PRINUM  := DEF_CURRENT^.PRINUM;
0559              INPUT_NUMBER    := DEFAULT_PRINUM;
0560
0561          END;
0562
0563      END;
0564
0565  END;
0566
0567  END;
0568
0569  END; { CHECK_DEFAULT }
```

```
0572      ( ++
0573
0574      DELETE_FDL_LINE -- Get rid of a line_object.
0575
0576      This routine lets the user find and remove a line_object from the Definition
0577      Linked List.
0578
0579      CALLING SEQUENCE:
0580
0581      DELETE_FDL_LINE;
0582
0583      INPUT PARAMETERS:
0584
0585      none
0586
0587      IMPLICIT INPUTS:
0588
0589      FULL_PROMPT
0590      ANSI_REVERSE
0591      TAB
0592      DEF_HEAD
0593      DEF_CURRENT
0594      SYS$INPUT:
0595
0596      OUTPUT PARAMETERS:
0597
0598      none
0599
0600      IMPLICIT OUTPUTS:
0601
0602      FDL_DEST
0603      DEF_CURRENT
0604      SYS$OUTPUT:
0605
0606      ROUTINES CALLED:
0607
0608      CLEAR
0609      ASK_DELETE_OPTION
0610      SHOW_CURRENT
0611      INCR_CURRENT
0612
0613      ROUTINE VALUE:
0614
0615      none
0616
0617      SIGNALS:
0618
0619      none
0620
0621      SIDE EFFECTS:
0622
0623      none
0624
0625      -- }
```

0627 PROCEDURE DELETE_FDL_LINE;

0628 VAR

```
0629     SAVE                : LINE_OBJECT;  
0630     DEF_REM_PRI         : ^LINE_OBJECT;  
0631     REMAINING_PRI      : BOOLEAN;  
0632     REMAINING_SEC      : BOOLEAN;  
0633     NO_MORE_PRI        : BOOLEAN;  
0634     FOUND_IT           : BOOLEAN;
```

0635 BEGIN

```
0636     { +  
0637     If the Definition Linked List is not empty, then do it, else skip it.  
0638     - }  
0639     IF DEF_HEAD <> DEF_TAIL THEN
```

0640 BEGIN

```
0641     SAVE.STRING          := NULL_STRING;  
0642     TEST.STRING          := NULL_STRING;
```

0643 CHECK_DEFAULT;

```
0644     { +  
0645     These routines will only return if an existing line_object has been given.  
0646     If 'EXTANT_ONLY' is specified.  
0647     - }
```

```
0648     FULL_CHOICE          := FALSE;  
0649     QUERY (EDF$K_TEST_PRIMARY);
```

0650 NO_MORE_PRI := FALSE;

```
0651     FULL_CHOICE          := FALSE;  
0652     ASK_TEST_SECONDARY;
```

```
0653     { +  
0654     Remember which primary it was.  
0655     - }
```

0656 SAVE := TEST;

```
0657     FOUND_IT             := FIND_OBJECT (  
0658         TEST.OBJECT_TYPE, TEST.PRIMARY,  
0659         TEST.PRNUM, TEST.SECONDARY, TEST.SECNUM  
0660     );
```

```
0661     { +  
0662     Setup to display definition on the terminal.  
0663     - }
```

```
0664     OPEN (FDL_DEST, SYSS$OUTPUT_NAME, NEW, RECORD_LENGTH := 252);  
0665     REWRITE (FDL_DEST);
```

0666 SHOW_CUR_PRI_SEC (TRUE);

0667 CLOSE (FDL_DEST);

0668 QUERY (EDF\$K_RETURN) ;
0669

```

DELETE_CURRENT;

IF TEST.PRIMARY <> TITLE THEN
BEGIN
  ( +
  Look through the list to see what remains of this primary.
  - )
  REMAINING_PRI      := FALSE;
  REMAINING_SEC      := FALSE;

  DEF_CURRENT := DEF_HEAD;

  REPEAT
    IF (
      (DEF_CURRENT^.PRIMARY = SAVE.PRIMARY)
      AND
      (DEF_CURRENT^.PRINUM = SAVE.PRINUM)
    ) THEN
      BEGIN
        IF DEF_CURRENT^.OBJECT_TYPE = PRI THEN
          BEGIN
            REMAINING_PRI := TRUE;
            DEF_REM_PRI   := DEF_CURRENT;
          END
        ELSE IF DEF_CURRENT^.OBJECT_TYPE = SEC THEN
          REMAINING_SEC := TRUE;
        END;
        INCR_CURRENT;
      UNTIL (REMAINING_PRI AND REMAINING_SEC) OR (DEF_CURRENT = NIL);
      IF (
        (REMAINING_PRI)
        AND
        (NOT REMAINING_SEC)
      ) THEN
        BEGIN
          WRITELN (CRLF,SHIFT,ANSI REVERSE,
            ' No more Secondaries with this Primary, deleting Primary. ',
            ANSI_RESET);
          DEF_CURRENT := DEF_REM_PRI;
        END
      END
    END
  END
END

```

EDFFUNCS
V04-000

Source Listing

M 15
16-Sep-1984 01:17:14
5-Sep-1984 13:37:08

VAX-11 Pascal V2.4-277 Page 17
DISK\$VMSMASTER:[EDF.SRC]EDFFUNCS.PAS;1 (10)

```
0741      DELETE CURRENT;  
0742      NO_MORE_PRI                := TRUE;  
0743  
0744      LIB$WAIT (3.0);  
0745  
0746      END  
0747  
0748      ELSE IF (  
0749      (NOT REMAINING_PRI)  
0750      AND  
0751      (REMAINING_SEC)  
0752      ) THEN  
0753  
0754          { NULL-STATEMENT }  
0755  
0756      ELSE IF (  
0757      (NOT REMAINING_PRI)  
0758      AND  
0759      (NOT REMAINING_SEC)  
0760      ) THEN  
0761  
0762      BEGIN  
0763  
0764          WRITELN (CRLF,SHIFT,ANSI_REVERSE,  
0765          ' This Primary Section has now been entirely Deleted. ',  
0766          ANSI_RESET);  
0767          NO_MORE_PRI                := TRUE;  
0768  
0769          LIB$WAIT (2.0);  
0770  
0771      END  
0772  
0773      ELSE IF (  
0774      (REMAINING_PRI)  
0775      AND  
0776      (REMAINING_SEC)  
0777      ) THEN  
0778  
0779      BEGIN  
0780  
0781          CLEAR (SCREEN);  
0782  
0783          WRITELN (SHIFT,TAB,TAB,ANSI_REVERSE,  
0784          ' Resulting Primary Section ',  
0785          ANSI_RESET,CRLF);  
0786  
0787          OPEN      (FDL_DEST,SYSS$OUTPUT_NAME,NEW,  
0788          RECORD_LENGTH := 252);  
0789          REWRITE (FDL_DEST);  
0790  
0791          SHOW_PRIMARY_SECTION (SAVE);  
0792  
0793          CLOSE (FDL_DEST);  
0794  
0795      END;  
0796  
0797      TEST.PRIMARY                := SAVE.PRIMARY;
```

EDFFUNCS
V04-000

Source Listing

N 15
16-Sep-1984 01:17:14
5-Sep-1984 13:37:08

VAX-11 Pascal V2.4-277
DISK\$VMSMASTER:[EDF.SRC]EDFFUNCS.PAS;1 (10) Page 18

```
0798      TEST.PRINUM      := SAVE.PRINUM;
0799
0800      IF NOT NO_MORE_PRI THEN
0801          QUERY (EDF$K_RETURN);
0802
0803      END;      { IF TEST.PRIMARY <> TITLE }
0804
0805      END      { IF TRUE DEF_HEAD <> DEF_TAIL }
0806
0807      ELSE
0808
0809      BEGIN
0810
0811          WRITELN (
0812              SHIFT,ANSI_REVERSE,' The Current Definition is Empty. ',ANSI_RESET);
0813
0814          LIB$WAIT (3.0);
0815
0816      END;      { IF FALSE DEF_HEAD <> DEF_TAIL }
0817
0818      END;      { DELETE_FDL_LINE }
0819
```

```
0821 { ++
0822
0823 MODIFY_FDL_LINE -- Modify an extant line_object.
0824
0825 This routine lets the user view and change the contents of a particular
0826 line_object in the Definition Linked List.
0827
0828 CALLING SEQUENCE:
0829
0830 MODIFY_FDL_LINE:
0831
0832 INPUT PARAMETERS:
0833
0834 none
0835
0836 IMPLICIT INPUTS:
0837
0838 SYSS$INPUT:
0839 The Definition Linked List
0840
0841 OUTPUT PARAMETERS:
0842
0843 none
0844
0845 IMPLICIT OUTPUTS:
0846
0847 SYSS$OUTPUT:
0848 The Definition Linked List
0849
0850 ROUTINES CALLED:
0851
0852 none
0853
0854 ROUTINE VALUE:
0855
0856 none
0857
0858 SIGNALS:
0859
0860 none
0861
0862 SIDE EFFECTS:
0863
0864 none
0865
0866 -- }
```

```
0868 PROCEDURE MODIFY_FDL_LINE;
0869
0870 VAR
0871     SAVE          : LINE_OBJECT;
0872     FOUND_IT      : BOOLEAN;
0873
0874 BEGIN
0875
0876     { +
0877     If the Definition Linked List is not empty, then do it, else skip it.
0878     - }
0879     IF DEF_HEAD <> DEF_TAIL THEN
0880
0881     BEGIN
0882
0883         SAVE.STRING          := NULL_STRING;
0884         SAVE.COMMENT         := NULL_STRING;
0885         TEST.STRING          := NULL_STRING;
0886         TEST.COMMENT         := NULL_STRING;
0887
0888         CHECK_DEFAULT;
0889
0890         { +
0891         These routines will only return if an existing line_object has been given.
0892         If 'EXTANT_ONLY' is specified.
0893         - }
0894         FULL_CHOICE          := FALSE;
0895         QUERY (EDF$K_TEST_PRIMARY);
0896
0897         FULL_CHOICE          := FALSE;
0898         ASK_TEST_SECONDARY;
0899
0900         FOUND_IT             := FIND_OBJECT (
0901                                     TEST.OBJECT_TYPE, TEST.PRIMARY, TEST.PRINUM,
0902                                     TEST.SECONDARY, TEST.SECNUM
0903                                     );
0904
0905         SAVE                  := DEF_CURRENT^;
0906
0907         { +
0908         Setup to display definition on the terminal.
0909         - }
0910         OPEN (FDL_DEST, SYSS$OUTPUT_NAME, NEW, RECORD_LENGTH := 252);
0911         REWRITE (FDL_DEST);
0912
0913         SHOW_CUR_PRI_SEC (TRUE);
0914
0915         CLOSE (FDL_DEST);
0916
0917         TEST := SAVE;
0918
0919         ASK_TEST_SECONDARY_VALUE;
0920
0921         MAKE_SCRATCH;
0922
0923         DEF_SCRATCH^ := TEST;
0924
```

```
0925 IF DEF_SCRATCH^.PRIMARY = TITLE THEN
0926
0927     DEF_SCRATCH^.OBJECT_TYPE := PRI
0928
0929 ELSE
0930
0931     DEF_SCRATCH^.OBJECT_TYPE := SEC;
0932
0933 INSERT_IN_ORDER (REPLACE_OBJ);
0934
0935 CLEAR (SCREEN);
0936
0937 WRITELN (SHIFT,TAB,TAB,ANSI_REVERSE,
0938 ' Resulting Primary Section ',
0939 ANSI_RESET,CRLF);
0940
0941 OPEN (FDL_DEST,SYSS$OUTPUT_NAME,NEW,RECORD_LENGTH := 252);
0942 REWRITE (FDL_DEST);
0943
0944 SHOW_PRIMARY_SECTION (SAVE);
0945
0946 CLOSE (FDL_DEST);
0947
0948 TEST := SAVE;
0949
0950 QUERY (EDF$K_RETURN);
0951
0952 END { IF TRUE DEF_HEAD <> DEF_TAIL }
0953
0954 ELSE
0955
0956 BEGIN
0957
0958     WRITELN (
0959     SHIFT,ANSI_REVERSE,' The Current Definition is Empty. ',ANSI_RESET);
0960
0961     LIB$WAIT (3.0);
0962
0963 END; { IF FALSE DEF_HEAD <> DEF_TAIL }
0964
0965 END; { MODIFY_FDL_LINE }
```

```
0967 { ++
0968
0969 HELP_PROC -- Prompt for help and process it.
0970
0971 This routine interfaces to the LBR$OUTPUT_HELP routine to access the
0972 help library.
0973
0974 CALLING SEQUENCE:
0975
0976 HELP_PROC;
0977
0978 INPUT PARAMETERS:
0979
0980 none
0981
0982 IMPLICIT INPUTS:
0983
0984 The help library: SYSS$LIBRARY:EDF.HLB
0985
0986 OUTPUT PARAMETERS:
0987
0988 none
0989
0990 IMPLICIT OUTPUTS:
0991
0992 SYSS$OUTPUT: (through lib$put_output)
0993
0994 ROUTINES CALLED:
0995
0996 LBR$OUTPUT_HELP
0997
0998 ROUTINE VALUE:
0999
1000 none
1001
1002 SIGNALS:
1003
1004 none
1005
1006 SIDE EFFECTS:
1007
1008 none
1009
1010 -- }
```

```
1012  PROCEDURE HELP_PROC;  
1013  
1014  BEGIN  
1015      { +  
1016      Call the Librarian's help routine that will prompt the user for any  
1017      additional information.  
1018      - }  
1019      ISTATUS      := LBR$OUTPUT_HELP (   
1020                                     IADDRESS (LIB$PUT_OUTPUT),  
1021                                     LINE_WIDTH,  
1022                                     0,  
1023                                     EDFHLP_STRING,  
1024                                     0,  
1025                                     IADDRESS (LIB$GET_INPUT)  
1026                                     );  
1027  
1028      { +  
1029      Show what the problem is.  
1030      - }  
1031      IF NOT ODD (ISTATUS) THEN  
1032          LIB$SIGNAL (ISTATUS,0,0,0);  
1033  
1034  END;      { HELP_PROC }  
1035  
1036
```

```
1038 { ++
1039
1040 VERIFY_ISAM_DEFINITION -- Check the linked list.
1041
1042 This routine verifies that the FDL definition is there and is indexed.
1043
1044 CALLING SEQUENCE:
1045
1046 boolean := VERIFY_ISAM_DEFINITION;
1047
1048 INPUT PARAMETERS:
1049
1050 none
1051
1052 IMPLICIT INPUTS:
1053
1054 none
1055
1056 OUTPUT PARAMETERS:
1057
1058 none
1059
1060 IMPLICIT OUTPUTS:
1061
1062 none
1063
1064 ROUTINES CALLED:
1065
1066 none
1067
1068 ROUTINE VALUE:
1069
1070 true or false depending upon the checking
1071
1072 SIGNALS:
1073
1074 none
1075
1076 SIDE EFFECTS:
1077
1078 none
1079
1080 -- }
```

```
1082 FUNCTION VERIFY_ISAM_DEFINITION : BOOLEAN;
1083
1084 VAR
1085     NON_EMPTY    : BOOLEAN;
1086     ISAM_FDL     : BOOLEAN;
1087
1088 BEGIN
1089     NON_EMPTY    := FALSE;
1090     ISAM_FDL     := FALSE;
1091
1092     { +
1093     Check for a definition that has more than an Ident.
1094     - }
1095     IF (
1096         (DEF_HEAD <> DEF_TAIL)
1097         OR
1098         (DEF_HEAD^.PRIMARY <> IDENT)
1099     ) THEN
1100
1101     BEGIN
1102         NON_EMPTY    := TRUE;
1103
1104         { +
1105         See what type of file the definition is now.
1106         1st, find the line_object that tells that.
1107         - }
1108         IF FIND_OBJECT (SEC,FILES$,0,ORGANIZATION,0) THEN
1109
1110         BEGIN
1111             IF DEF_CURRENT^.QUALIFIER = FDL$C_IDX THEN
1112
1113             ISAM_FDL    := TRUE;
1114
1115         END;    { IF TRUE FIND_OBJECT () }
1116
1117         IF NOT ISAM_FDL THEN
1118
1119         BEGIN
1120             WRITELN (SHIFT,ANSI_REVERSE,
1121                     ' The current file organization is not Indexed. ',
1122                     ANSI_RESET);
1123             LIB$WAIT (3.0);
1124
1125         END;    { IF FALSE ISAM_FDL }
1126
1127     END { IF TRUE (DEF_HEAD <> DEF_TAIL) OR (DEF_HEAD^.PRIMARY <> IDENT) }
1128
1129     ELSE
1130
1131     IF NOT AUTO_TUNE THEN
1132     BEGIN
```

EDFFUNCS
V04-000

Source Listing

I 16
16-Sep-1984 01:17:14
5-Sep-1984 13:37:08

VAX-11 Pascal V2.4-277
DISK\$VMSMASTER:[EDF.SRC]EDFFUNCS.PAS;1 (16) Page 26

```
1139      ( +
1140      Slap the user's wrist.
1141      - )
1142      WRITELN (SHIFT,ANSI_REVERSE,
1143      ' The current FDL Definition is empty. ',
1144      ANSI_RESET);
1145
1146      LIB$WAIT (3.0);
1147
1148      END
1149  ELSE
1150      BEGIN
1151      LIB$SIGNAL (EDF$_INSFANL,0,0,0); {no definition like above}
1152      END;
1153
1154      ( +
1155      We must have something, and that something must be indexed.
1156      - )
1157      VERIFY_ISAM_DEFINITION      := (NON_EMPTY AND ISAM_FDL);
1158
1159  END;      { VERIFY_ISAM_DEFINITION }
```

EDFFUNCS
V04-000

Source Listing

J 16
16-Sep-1984 01:17:14
5-Sep-1984 13:37:08

VAX-11 Pascal V2.4-277
DISK\$VMSMASTER:[EDF.SRC]EDFFUNCS.PAS;1 (17) Page 27

```
1162 { ++
1163
1164 REDESIGN_SCRIPT_PROC -- Redesign a definition.
1165
1166 This routine allows old definitions to done over.
1167
1168 CALLING SEQUENCE:
1169
1170 REDESIGN_SCRIPT_PROC;
1171
1172 INPUT PARAMETERS:
1173
1174 none
1175
1176 IMPLICIT INPUTS:
1177
1178 none
1179
1180 OUTPUT PARAMETERS:
1181
1182 none
1183
1184 IMPLICIT OUTPUTS:
1185
1186 none
1187
1188 ROUTINES CALLED:
1189
1190 INDEXED_DESIGN
1191
1192 ROUTINE VALUE:
1193
1194 none
1195
1196 SIGNALS:
1197
1198 none
1199
1200 SIDE EFFECTS:
1201
1202 none
1203
1204 -- }
```

EDFFUNCS
V04-000

Source Listing

K 16
16-Sep-1984 01:17:14
5-Sep-1984 13:37:08

VAX-11 Pascal V2.4-277
DISK\$VMSMASTER:LEDFFUNCS.PAS;1 (18) Page 28

```
1206 PROCEDURE REDESIGN_SCRIPT_PROC;  
1207  
1208 BEGIN  
1209  
1210     IF VERIFY_ISAM_DEFINITION THEN  
1211         INDEXED_DESIGN (TRUE,FALSE);  
1212  
1213  
1214 END;    { REDESIGN_SCRIPT_PROC }
```

EDFFUNCS
V04-000

Source Listing

L 16
16-Sep-1984 01:17:14
5-Sep-1984 13:37:08

VAX-11 Pascal V2.4-277
DISK\$VMSMASTER:[EDF.SRC]EDFFUNCS.PAS;1 (19) Page 29

```
1216 { ++
1217
1218 ADD_KEY_SCRIPT_PROC -- Define a new key.
1219
1220 This routine allows new keys to be added to the definition.
1221
1222 CALLING SEQUENCE:
1223
1224 ADD_KEY_SCRIPT_PROC;
1225
1226 INPUT PARAMETERS:
1227
1228 none
1229
1230 IMPLICIT INPUTS:
1231
1232 none
1233
1234 OUTPUT PARAMETERS:
1235
1236 none
1237
1238 IMPLICIT OUTPUTS:
1239
1240 none
1241
1242 ROUTINES CALLED:
1243
1244 REDESIGN_FDL
1245
1246 ROUTINE VALUE:
1247
1248 none
1249
1250 SIGNALS:
1251
1252 none
1253
1254 SIDE EFFECTS:
1255
1256 none
1257
1258 -- }
```

EDFFUNCS
V04-000

Source Listing

M 16
16-Sep-1984 01:17:14
5-Sep-1984 13:37:08

VAX-11 Pascal V2.4-277
DISK\$VMSMASTER:[EDF.SRC]EDFFUNCS.PAS;1 (20) Page 30

```
1260 PROCEDURE ADD_KEY_SCRIPT_PROC;  
1261  
1262 BEGIN  
1263  
1264     IF VERIFY_ISAM_DEFINITION THEN  
1265  
1266     BEGIN  
1267  
1268         { +  
1269         See what we have already.  
1270         - }  
1271         SCAN_DEFINITION (FALSE);  
1272  
1273         { +  
1274         Set the key that we have to redesign.  
1275         - }  
1276         IDATA[EDF$K_ACTIVE_KEY] := HIGH_KEY;  
1277  
1278         IF FOUND_0 THEN  
1279  
1280             IDATA[EDF$K_ACTIVE_KEY] := IDATA[EDF$K_ACTIVE_KEY] + 1;  
1281  
1282         { +  
1283         Go model and select those parameters.  
1284         - }  
1285         INDEXED_DESIGN (TRUE,TRUE);  
1286  
1287     END;      { IF TRUE VERIFY_ISAM_DEFINITION }  
1288  
1289 END;      { ADD_KEY_SCRIPT_PROC }
```

```
1291      { ++
1292
1293      DELETE_KEY_SCRIPT_PROC -- Remove a key definition from the Link List.
1294
1295      This routine allows key definitions to be removed - along with the
1296      accompanying area proposals.
1297
1298      CALLING SEQUENCE:
1299
1300      DELETE_KEY_SCRIPT_PROC:
1301
1302      INPUT PARAMETERS:
1303
1304      none
1305
1306      IMPLICIT INPUTS:
1307
1308      none
1309
1310      OUTPUT PARAMETERS:
1311
1312      none
1313
1314      IMPLICIT OUTPUTS:
1315
1316      none
1317
1318      ROUTINES CALLED:
1319
1320      INDEXED_DESIGN
1321
1322      ROUTINE VALUE:
1323
1324      none
1325
1326      SIGNALS:
1327
1328      none
1329
1330      SIDE EFFECTS:
1331
1332      none
1333      -- }
1334
```

```
1336 PROCEDURE DELETE_KEY_SCRIPT_PROC;  
1337  
1338   PROCEDURE DELETE_SECTION (SECTION : PRIMARY_TYPE; SECT_NUM : INTEGER);  
1339  
1340     BEGIN  
1341  
1342       IF FIND_OBJECT (PRI,SECTION,SECT_NUM,DUMMY_SECONDARY$,0) THEN  
1343  
1344         BEGIN  
1345  
1346           WRITELN (SHIFT,'Deleting '  
1347             SECTION:PRIMARY_WIDTH[SECTION],SECT_NUM:3,' primary section.');
```

1348 QUERY (EDFSK RETURN);
1349 DELETE_PRIMARY_SECTION (SECTION,SECT_NUM);
1350
1351 END; { IF TRUE FIND_OBJECT () }
1352
1353 END; { DELETE_AREA }

1354
1355 VAR
1356 LO_AREA : INTEGER;
1357 L1_AREA : INTEGER;
1358 LX_AREA : INTEGER;
1359
1360 BEGIN
1361
1362 IF VERIFY_ISAM_DEFINITION THEN
1363
1364 BEGIN
1365
1366 (+
1367 See what we have.
1368 -)
1369 SCAN_DEFINITION (TRUE);
1370
1371 IF HIGH_KEY <> 0 THEN
1372
1373 BEGIN
1374
1375 (+
1376 See which areas are used by this key.
1377 -)
1378 IF FIND_OBJECT (SEC,KEY,HIGH_KEY,DATA_AREA,0) THEN
1379
1380 LO_AREA := DEF_CURRENT^.NUMBER
1381
1382 ELSE
1383
1384 LO_AREA := -1;
1385
1386 IF FIND_OBJECT (SEC,KEY,HIGH_KEY,LEVEL1_INDEX_AREA,0) THEN
1387
1388 L1_AREA := DEF_CURRENT^.NUMBER
1389
1390 ELSE
1391
1392 L1_AREA := -1;

```
1393 IF FIND_OBJECT (SEC,KEY,HIGH_KEY,INDEX_AREA,0) THEN
1394
1395     LX_AREA      := DEF_CURRENT^.NUMBER
1396
1397 ELSE
1398
1399     LX_AREA      := -1;
1400
1401 { +
1402 Eliminate those areas that are also used by other keys.
1403 - }
1404
1405 DEF_CURRENT      := DEF_HEAD;
1406
1407 WITH DEF_CURRENT^ DO
1408
1409 BEGIN
1410
1411     REPEAT
1412
1413         IF (
1414             (PRIMARY = KEY)
1415             AND
1416             (PRINUM <> HIGH_KEY)
1417             AND
1418             (SECONDARY = DATA_AREA)
1419             AND
1420             (NUMBER = LO_AREA)
1421         ) THEN
1422
1423             LO_AREA := -1;
1424
1425         IF (
1426             (PRIMARY = KEY)
1427             AND
1428             (PRINUM <> HIGH_KEY)
1429             AND
1430             (SECONDARY = LEVEL1_INDEX_AREA)
1431             AND
1432             (NUMBER = L1_AREA)
1433         ) THEN
1434
1435             L1_AREA := -1;
1436
1437         IF (
1438             (PRIMARY = KEY)
1439             AND
1440             (PRINUM <> HIGH_KEY)
1441             AND
1442             (SECONDARY = INDEX_AREA)
1443             AND
1444             (NUMBER = LX_AREA)
1445         ) THEN
1446
1447             LX_AREA := -1;
1448
1449         INCR_CURRENT;
```

```
1450      UNTIL DEF_CURRENT = NIL;
1451
1452      END;      { DO }
1453
1454      { +
1455      Get rid of the key definition.
1456      - }
1457      DELETE_SECTION (KEY,HIGH_KEY);
1458
1459      { +
1460      Get rid of any now obsolete area definitions.
1461      - }
1462      IF NOT (LO_AREA < 0) THEN
1463          DELETE_SECTION (AREA,LO_AREA);
1464
1465      IF NOT (L1_AREA < 0) THEN
1466          DELETE_SECTION (AREA,L1_AREA);
1467
1468      IF NOT (LX_AREA < 0) THEN
1469          DELETE_SECTION (AREA,LX_AREA);
1470
1471      WRITELN (SHIFT,'End of Delete_Key_Indexed Script. ');
1472      QUERY (EDF$K_RETURN);
1473
1474      END
1475
1476      ELSE
1477
1478      BEGIN
1479          WRITELN (SHIFT,ANSI_REVERSE,
1480          ' This script will not delete the Primary Key. ',
1481          ANSI_RESET);
1482          LIB$WAIT (3.0);
1483
1484      END;
1485
1486      END;      { IF TRUE VERIFY_ISAM_DEFINITION }
1487
1488      END;      { DELETE_KEY_SCRIPT_PROC }
```

```
1496      ( ++
1497
1498      OPTIMIZE_SCRIPT_PROC -- Optimize extant definitions.
1499
1500      This routine allows old definitions to modified and optimized.
1501
1502      CALLING SEQUENCE:
1503
1504      OPTIMIZE_SCRIPT_PROC;
1505
1506      INPUT PARAMETERS:
1507
1508      none
1509
1510      IMPLICIT INPUTS:
1511
1512      none
1513
1514      OUTPUT PARAMETERS:
1515
1516      none
1517
1518      IMPLICIT OUTPUTS:
1519
1520      none
1521
1522      ROUTINES CALLED:
1523
1524      none
1525
1526      ROUTINE VALUE:
1527
1528      none
1529
1530      SIGNALS:
1531
1532      none
1533
1534      SIDE EFFECTS:
1535
1536      none
1537
1538      -- }
```

```
1540 PROCEDURE OPTIMIZE_SCRIPT_PROC;
1541
1542 VAR
1543     AN_KEY_FOUND      : BOOLEAN;
1544
1545 BEGIN
1546     IF NOT ANALYSIS_SPECIFIED THEN
1547     BEGIN
1548         IF NOT (AUTO_TUNE) THEN
1549             WRITELN (SHIFT,
1550                 'An Input Analysis File is necessary for Optimizing Keys.',
1551                 CRLF_SHIFT)
1552         ELSE
1553             ( + exit since nointerative and no analysis file
1554             - )
1555             LIB$STOP (EDF$_INSFANL,0,0,0);
1556
1557     END;
1558     INPUT_ANALYSIS_FILE;
1559     AN_KEY_FOUND      := FALSE;
1560     POINT_AT_ANALYSIS;
1561     DEF_CURRENT := DEF_HEAD;
1562     REPEAT
1563         IF DEF_CURRENT^.PRIMARY = ANALYSIS_OF_KEY THEN
1564             AN_KEY_FOUND      := TRUE;
1565             INCR_CURRENT;
1566     UNTIL (AN_KEY_FOUND = TRUE) OR (DEF_CURRENT = NIL);
1567     POINT_AT_DEFINITION;
1568     IF AN_KEY_FOUND THEN
1569     BEGIN
1570         OPTIMIZING      := TRUE;
1571         REDESIGN_SCRIPT_PROC;
1572     END
```

EDFFUNCS
V04-000

Source Listing

H 1
16-Sep-1984 01:17:14
5-Sep-1984 13:37:08

VAX-11 Pascal V2.4-277
DISK\$VMSMASTER:[EDF.SRC]EDFFUNCS.PAS;1 (24) Page 37

```
1597
1598     ELSE
1599
1600     BEGIN
1601
1602     IF NOT (AUTO_TUNE) THEN
1603     BEGIN
1604         WRITELN (SHIFT,
1605             'The Analysis file must contain ANALYSIS_OF_KEY primary sections. ');
1606         WRITELN (SHIFT,
1607             'The DCL command "ANALYZE/RMS_FILE/FDL" produces Analysis Files. ');
1608
1609         CLEAR (PAUSE);
1610
1611     END
1612     END;
1613
1614     OPTIMIZING := FALSE;
1615 END;    ( OPTIMIZE_SCRIPT_PROC )
```

```
1617 { ++
1618
1619 INVOKE_SCRIPT -- Start up a series of questions.
1620
1621 This routine dispatches to the script procedures.
1622
1623 CALLING SEQUENCE:
1624
1625 INVOKE_SCRIPT;
1626
1627 INPUT PARAMETERS:
1628
1629 none
1630
1631 IMPLICIT INPUTS:
1632
1633 IDATA[EDF$K_SCRIPT_OPTION]
1634 SYSSINPUT_ERROR
1635 SYSSINPUT;
1636
1637 OUTPUT PARAMETERS:
1638
1639 none
1640
1641 IMPLICIT OUTPUTS:
1642
1643 SYSSINPUT_ERROR
1644 TEMP_FULL_PROMPT
1645
1646 ROUTINES CALLED:
1647
1648 OPTIMIZE_SCRIPT_PROC
1649 DESIGN_SCRIPT_PROC
1650
1651 ROUTINE VALUE:
1652
1653 none
1654
1655 SIGNALS:
1656
1657
1658 SIDE EFFECTS:
1659
1660 none
1661
1662 -- }
```

```
1664 PROCEDURE INVOKE_SCRIPT;
1665
1666 BEGIN
1667     { +
1668     Reset so 1st (DCL) script only gets done once.
1669     - }
1670     IDATA[EDFSK_FIRST_SCRIPT] := EDFSK_ZERO_SCRIPT;
1671
1672     { +
1673     Prompt for the desired script if we don't already have one. (from DCL)
1674     - }
1675     IF IDATA[EDFSK_SCRIPT_OPTION] = EDFSK_ZERO_SCRIPT THEN
1676     BEGIN
1677         { +
1678         See which script the user wants.
1679         - }
1680         QUERY (EDFSK_SCRIPT_OPTION);
1681     END { IF TRUE IDATA[EDFSK_SCRIPT_OPTION] = EDFSK_ZERO_SCRIPT }
1682 ELSE
1683 BEGIN
1684     IF NOT AUTO_TUNE THEN
1685     BEGIN
1686         CLEAR (SCREEN);
1687         WRITE (SHIFT,TAB,TAB,ANSI_REVERSE);
1688         CASE IDATA[EDFSK_SCRIPT_OPTION] OF
1689             EDFSK_ADD_KEY_FDL : WRITE (' Add Key');
1690             EDFSK_DELETE_KEY_FDL : WRITE (' Delete Key');
1691             EDFSK_IDX_DESIGN_FDL : WRITE (' Indexed');
1692             EDFSK_SEQ_DESIGN_FDL : WRITE (' Sequential');
1693             EDFSK_REL_DESIGN_FDL : WRITE (' Relative');
1694             EDFSK_OPTIMIZE_FDL : WRITE (' Optimize');
1695             EDFSK_REDESIGN_FDL : WRITE (' Touchup');
1696         OTHERWISE
1697             { NULL-STATEMENT } ;
1698         END; { CASE }
1699         WRITELN (' Script ',ANSI_RESET,CRLF);
1700     END; { IF NOT AUTO_TUNE }
1701 END; { IF FALSE IDATA[EDFSK_SCRIPT_OPTION] = EDFSK_ZERO_SCRIPT }
1702 TAKE_DEFAULTS := TRUE;
```

```
1721
1722 CASE IDATA[EDF$K_SCRIPT_OPTION] OF
1723     EDF$K_IDX_DESIGN_FDL :
1724         BEGIN
1725             WARN_OF_ERASE;
1726             INIT_DEF;
1727             INDEXED_DESIGN (FALSE,FALSE);
1728         END;
1729     EDF$K_SEQ_DESIGN_FDL :
1730         BEGIN
1731             WARN_OF_ERASE;
1732             INIT_DEF;
1733             SEQ_REL_WORK;
1734             SEQ_DEF;
1735         END;
1736     EDF$K_REL_DESIGN_FDL :
1737         BEGIN
1738             WARN_OF_ERASE;
1739             INIT_DEF;
1740             SEQ_REL_WORK;
1741             REL_DEF;
1742         END;
1743     EDF$K_ADD_KEY_FDL :    ADD_KEY_SCRIPT_PROC;
1744     EDF$K_DELETE_KEY_FDL : DELETE_KEY_SCRIPT_PROC;
1745     EDF$K_OPTIMIZE_FDL :  OPTIMIZE_SCRIPT_PROC;
1746     EDF$K_REDESIGN_FDL :  REDESIGN_SCRIPT_PROC;
1747 OTHERWISE
1748     ( NULL-STATEMENT ) ;
1749 END;      ( CASE )
1750 TAKE_DEFAULTS      := FALSE;
1751 END;      ( INVOKE_SCRIPT )
1752
```

```
1774 { ++
1775
1776 SET_PROC -- Set the characteristics of the FDL Editor.
1777
1778 This routine asks which characteristics are to be set and sets them.
1779
1780 CALLING SEQUENCE:
1781
1782 SET_PROC;
1783
1784 INPUT PARAMETERS:
1785
1786 none
1787
1788 IMPLICIT INPUTS:
1789
1790 SYSS$INPUT_ERROR
1791 SYSS$INPUT;
1792
1793 OUTPUT PARAMETERS:
1794
1795 none
1796
1797 IMPLICIT OUTPUTS:
1798
1799 SYSS$INPUT_ERROR
1800
1801 ROUTINES CALLED:
1802
1803
1804 ROUTINE VALUE:
1805
1806 none
1807
1808 SIGNALS:
1809
1810 SIDE EFFECTS:
1811
1812 none
1813
1814 -- }
1815
```

```
1817 PROCEDURE SET_PROC;
1818
1819 BEGIN
1820     VISIBLE_QUESTION := TRUE;
1821
1822     QUERY (EDFSK_SET_FUNCTION);
1823
1824     CASE IDATA[EDFSK_SET_FUNCTION] OF
1825
1826         EDFSK_SET_DISPLAY :    QUERY (EDFSK_SURFACE_OPTION);
1827         EDFSK_SET_EMPHASIS :   QUERY (EDFSK_BUCKET_WEIGHT);
1828         EDFSK_SET_GRANULARITY : QUERY (EDFSK_GRANULARITY);
1829         EDFSK_SET_RESPONSES :  QUERY (EDFSK_RESPONSES);
1830         EDFSK_SET_PROMPTING :  QUERY (EDFSK_PROMPTING);
1831         EDFSK_SET_ANALYSIS :   QUERY (EDFSK_ANALYSIS);
1832         EDFSK_SET_OUTPUT :     QUERY (EDFSK_OUTPUT);
1833
1834         EDFSK_SET_NUMBER_KEYS :
1835
1836             BEGIN
1837
1838                 QUERY (EDFSK_NUMBER_KEYS);
1839                 NUMBER_KEYS_SET := TRUE;
1840
1841             END;
1842
1843     OTHERWISE
1844
1845         { NULL-STATEMENT } ;
1846
1847     END;      { CASE }
1848
1849     VISIBLE_QUESTION := FALSE;
1850
1851 END;      { SET_PROC }
1852
1853 END.
1854 { End of file: SRC$:EDFFUNCS.PAS }
1855
```

```
65 72 63 20 74 6F 6E 20 74 75 70 74 75 4F
74 6E 65 72 72 75 43 20 2D 20 64 65 74 61
6F 69 74 69 6E 69 66 65 44 20 4C 44 46 20
00 00 2E 79 74 70 6D 65 20 6E
69 72 50 20 67 6E 69 74 6C 75 73 65 52 20
00 20 6E 6F 69 74 63 65 53 20 79 72 61 6D
6E 6F 63 65 53 20 65 72 6F 6D 20 6F 4E 20
68 74 20 68 74 69 77 20 73 65 69 72 61 64
65 64 20 2C 79 72 61 6D 69 72 50 20 73 69
79 72 61 6D 69 72 50 20 67 6E 69 74 65 6C
00 00 20 2E
20 79 72 61 6D 69 72 50 20 73 69 68 54 20
6F 6E 20 73 61 68 20 6E 6F 69 74 63 65 53
6C 65 72 69 74 6E 65 20 6E 65 65 62 20 77
00 00 00 20 2E 64 65 74 65 6C 65 44 20 79
69 72 50 20 67 6E 69 74 6C 75 73 65 52 20
00 20 6E 6F 69 74 63 65 53 20 79 72 61 6D
44 20 74 6E 65 72 75 43 20 65 68 54 20
45 20 73 69 20 6E 6F 69 74 69 6E 69 66 65
00 00 20 2E 79 74 70 6D
69 72 50 20 67 6E 69 74 6C 75 73 65 52 20
00 20 6E 6F 69 74 63 65 53 20 79 72 61 6D
44 20 74 6E 65 72 75 43 20 65 68 54 20
45 20 73 69 20 6E 6F 69 74 69 6E 69 66 65
00 00 20 2E 79 74 70 6D
66 20 74 6E 65 72 72 75 63 20 65 68 54 20
69 74 61 7A 69 6E 61 67 72 6F 20 65 6C 69
65 64 6E 49 20 74 6F 6E 20 73 69 20 6E 6F
00 20 2E 64 65 78
46 20 74 6E 65 72 72 75 63 20 65 68 54 20
20 6E 6F 69 74 69 6E 69 66 65 44 20 4C 44
00 00 20 2E 79 74 70 6D 65 20 73 69
5F 65 74 65 6C 65 44 20 66 6F 20 64 6E 45
63 53 20 64 65 78 65 64 6E 49 5F 79 65 48
00 00 00 2E 74 70 69 72
77 20 74 70 69 72 63 73 20 73 69 68 54 20
65 74 65 6C 65 64 20 74 6F 6E 20 6C 6C 69
48 20 79 72 61 6D 69 72 50 20 65 68 74 20
00 00 20 2E 79 65
00 00 00 20 67 6E 69 74 65 6C 65 44
0000006B 00000064 00000055 00000010 00000048
0000009D 00000095 00000090 00000080 0000006F
000000BA 000000B6 000000AE 000000A8 000000A2
000000D1 000000CA 000000C2
59 45 50 59 54 5F 59 52 41 4D 49 52 50 5F 59 4D 4D 55 44 0E
24
53 53 45 43 43 41 06
4C 43 41 03
41 5F 46 4F 5F 53 49 53 59 4C 41 4E 41 10
41 45 52
48 5F 46 4F 5F 53 49 53 59 4C 41 4E 41 0F
```

```
00000 .PSECT $CODE,PIC,CON,REL,LCL,SHR,EXE,RD,NOWRT,2
00000 C.AAA: .ASCII \Output not created - Current FDL Definit\
0000E \ion empty.\<0><0>
0001C
0002A
00034 C.AAB: .ASCII \ Resulting Primary Section \<0>
00042
00050 C.AAC: .ASCII \ No more Secondaries with this Primary, \-
0005E \deleting Primary. \<0><0>
0006C
0007A
00088
0008C C.AAD: .ASCII \ This Primary Section has now been entir\
0009A \ely Deleted. \<0><0><0>
000A8
000B6
000C4 C.AAE: .ASCII \ Resulting Primary Section \<0>
000D2
000E0 C.AAF: .ASCII \ The Current Definition is Empty. \<0><0>
000EE
000FC
00104 C.AAG: .ASCII \ Resulting Primary Section \<0>
00112
00120 C.AAH: .ASCII \ The Current Definition is Empty. \<0><0>
0012E
0013C
00144 C.AAI: .ASCII \ The current file organization is not In\
00152 \dexed. \<0>
00160
0016E
00174 C.AAJ: .ASCII \ The current FDL Definition is empty. \-
00182 <0><0>
00190
0019C C.AAK: .ASCII \End of Delete_Key_Indexed Script.\<0>-
001AA <0><0>
001B8
001C0 C.AAL: .ASCII \ This script will not delete the Primary\
001CE \ Key. \<0><0>
001DC
001EA
001F0 C.AAM: .ASCII \Deleting \<0><0><0>
001FC C.AAN: .LONG 72,16,85,100,107,111,128,144,149,157,162,-
00210 168,174,182,186,194,202,209
00224
00238
00244 .ASCII <12>\PRIMARY TYPE\
00251 .ASCII <14>\DUMMY_PRIMARY$
0025F
00260 .ASCII <6>\ACCESS\
00267 .ASCII <3>\ACL\
0026B .ASCII <16>\ANALYSIS_OF_AREA\
00279
0027C .ASCII <15>\ANALYSIS_OF_KEY\
```

```

54 43 45 41 45 52 41 04
      4E 4E 4F 43 07
      45 54 41 44 04
      24 45 4C 49 46 05
      54 4E 45 44 49 05
4C 41 4E 52 55 4F 4A 07
      59 45 4B 03
24 44 52 4F 43 45 52 07
47 4E 49 52 41 48 53 07
      4D 45 54 53 59 53 06
      00 45 4C 54 49 54 05
69 74 63 65 73 20 79 72 61 6D 69 72 70 20
      00 00 00 2E 6E 6F
79 6C 61 6E 41 20 74 75 70 6E 49 20 6E 41
65 6E 20 73 69 20 65 6C 69 46 20 73 69 73
70 4F 20 72 6F 66 20 79 72 61 73 73 65 63
2E 73 79 65 4B 20 67 6E 69 7A 69 6D 69 74
46 20 73 69 73 79 6C 61 6E 41 20 65 68 54
61 74 6E 6F 63 20 74 73 75 6D 20 65 6C 69
46 4F 5F 53 49 53 59 4C 41 4E 41 20 6E 69
73 20 79 72 61 6D 69 72 70 20 59 45 4B 5F
      2E 73 6E 6F 69 74 63 65
6E 61 6D 6D 6F 63 20 4C 43 44 20 65 68 54
53 4D 52 2F 45 5A 59 4C 41 4E 41 22 20 64
6F 72 70 20 22 4C 44 46 2F 45 4C 49 46 5F
73 69 73 79 6C 61 6E 41 20 73 65 63 75 64
      00 2E 73 65 6C 69 46 20
      79 65 4B 5F 64 64 41 20
      5F 65 74 65 6C 65 44 20
      64 65 78 65 64 6E 49 20
      00 6C 61 69 74 6E 65 75 71 65 53 20
      00 00 00 65 76 69 74 61 6C 65 52 20
      00 00 00 65 7A 69 6D 69 74 70 4F 20
      70 75 68 63 75 6F 54 20
      20 74 70 69 72 63 53 20
```

```

0028A
0028C .ASCII <4>\AREA\
00291 .ASCII <7>\CONNECT\
00299 .ASCII <4>\DATE\
0029E .ASCII <5>\FILES\
002A4 .ASCII <5>\IDENT\
002AA .ASCII <7>\JOURNAL\
002B2 .ASCII <3>\KEY\
002B6 .ASCII <7>\RECORDS\
002BE .ASCII <7>\SHARING\
002C6 .ASCII <6>\SYSTEM\
002CD .ASCII <5>\TITLE\<0>
002D4 C.AAO: .ASCII \ primary section.\<0><0><0>
002E2
002E8 C.AAP: .ASCII \An Input Analysis File is necessary for \-
002F6 \Optimizing Keys.\
00304
00312
00320 C.AAQ: .ASCII \The Analysis File must contain ANALYSIS_\-
0032E \OF_KEY primary sections.\
0033C
0034A
00358
00360 C.AAR: .ASCII \The DCL command "ANALYZE/RMS_FILE/FDL" p\-
0036E \roduces Analysis Files.\<0>
0037C
0038A
00398
003A0 C.AAS: .ASCII \ Add_Key\
003A8 C.AAT: .ASCII \ Delete_Key\<0>
003B4 C.AAU: .ASCII \ Indexed\
003BC C.AAV: .ASCII \ Sequential\<0>
003C8 C.AAW: .ASCII \ Relative\<0><0><0>
003D4 C.AAX: .ASCII \ Optimize\<0><0><0>
003E0 C.AAY: .ASCII \ Touchup\
003E8 C.AAZ: .ASCII \ Script \
```

```

00000000G SE FEFO CE 0000 00000 CREATE_NEW_FDL: ; 0159
      EF 00000000G EF 9E 00002 .WORD ^M<>
      03 13 00007 MOVAB -272(SP),SP
      0000V 31 00012 CMPL DEF_HEAD,DEF_TAIL ; 0172
      00000000G EF 94 00014 BEQL +3-
      00000000G EF 9F 00017 BRW 2$
      00000000G EF 02 0001D CLRB FILE_CREATED ; 0176
      00000000G EF 9F 00023 PUSHAB CRLF ; 0178
      00000000G EF 9F 00025 PUSHL #2
      00000000G EF 03 FB 0002B PUSHAB PASS$V_OUTPUT
      00000000G EF 04 9F 00032 CALLS #3,PASS$WRITE_STRING
      00000000G EF 03 DD 00038 PUSHAB SHIFT
      00000000G EF 01 9F 0003A PUSHL #4
      00000000G EF 03 FB 00040 PUSHAB PASS$V_OUTPUT
      00000000G EF 01 DD 00047 CALLS #3,PASS$WRITE_STRING
      7E 00000000G EF 9A 00049 PUSHL #1
      00000000G EF 9F 00050 MOVZBL CONTROL_G,-(SP)
      00000000G EF 03 FB 00056 PUSHAB PASS$V_OUTPUT
      EF 9F 0005D CALLS #3,PASS$WRITE_CHAR
      PUSHAB ANSI_REVERSE
```

			04	DD	00063	PUSHL	#4		
			EF	9F	00065	PUSHAB	PASS\$V_OUTPUT		
00000000G	EF		03	FB	00068	CALLS	#3,PASS\$WRITE_STRING		
		FFFFFB98	EF	9F	00072	PUSHAB	C.AAA		
			32	DD	00078	PUSHL	#50		
00000000G	EF		EF	9F	0007A	PUSHAB	PASS\$V_OUTPUT		
		00000000G	03	FB	00080	CALLS	#3,PASS\$WRITE_STRING		
		00000000G	EF	9F	00087	PUSHAB	ANSI_RESET		
			04	DD	0008D	PUSHL	#4		
00000000G	EF		EF	9F	0008F	PUSHAB	PASS\$V_OUTPUT		
		00000000G	03	FB	00095	CALLS	#3,PASS\$WRITE_STRING		
00000000G	EF		EF	9F	0009C	PUSHAB	PASS\$V_OUTPUT		
			01	FB	000A2	CALLS	#1,PASS\$WRITELN2		
		0000V	31	000A9	BRW	13\$			
00000000G	EF	00000000G	EF	7D	000AC	MOVQ	NULL_STRING,RES_OUTPUT_FILENAME_DESC		: 0187
		000000FF	8F	DD	000B7	PUSHL	#255		: 0188
00000000G	EF		01	FB	000BD	CALLS	#1,PASS\$NEW2		
00000004G	EF		50	DD	000C4	MOVL	R0,RES_OUTPUT_FILENAME_DESC+4		
00000000G	EF	FF	8F	9B	000CB	MOVZBW	#255,RES_OUTPUT_FILENAME_DESC		: 0189
00000000G	EF		01	FO	000D3	INSV	#1,#0,#1,FLAGS		: 0190
			00	FO	000DC	INSV	#0,#4,#1,FLAGS		: 0191
			EF	9F	000E5	PUSHAB	FLAGS		: 0193
			AD	9F	000EB	PUSHAB	FID_BLOCK		
			EF	9F	000EE	PUSHAB	RES_OUTPUT_FILENAME_DESC		
			EF	9F	000F4	PUSHAB	DEFAULT_FILENAME_DESC		
			EF	9F	000FA	PUSHAB	OUTPUT_FILENAME_DESC		
			EF	9F	00100	PUSHAB	NL_DEV_DESC		
00000000G	EF		06	FB	00106	CALLS	#6,FDL\$CREATE		
00000000G	EF		50	DD	0010D	MOVL	R0,ISTATUS		
			03	EF	EB	00114	BLBS	ISTATUS,..+3	: 0202
			0000V	31	0011B	BRW	13\$		
			EF	94	0011E	CLRB	DEST_IS_TERMINAL		: 0209
			01	DD	00124	MOVL	#1,R0		: 0215
			50	DD	00127	MOVL	R0,I		
51 00000000G	EF		10	ED	0012A	CMPZV	#0,#16,RES_OUTPUT_FILENAME_DESC,I		: 0217
			00V	18	00133	BGEQ	6\$		
		FEF0 CD41	20	90	00135	MOVB	#32,TEMP_STRING255-1[I]		: 0219
			00V	11	0013B	BRB	7\$		
			EF	DD	0013D	MOVL	RES_OUTPUT_FILENAME_DESC+4,R12		: 0223
		5C 00000004G	EF	90	00144	MOVB	-1(R12)[I],TEMP_STRING255-1[I]		
		FEF0 CD41	FF AC41	F3	0014C	AOBLEQ	#255,R0,4\$		
D3		50 000000FF	8F	ED	00154	BBS	#0,AUTO_TUNE,9\$: 0230
00V00000000G	EF		00	DD	0015C	PUSHL	#252		: 0234
		000000FC	8F	DD	00162	PUSHL	#7		
			07	DD	00164	PUSHL	#4		
			04	DD	00166	PUSHAB	SY\$OUTPUT_NAME		
			0B	DD	0016C	PUSHL	#11		
			01	DD	0016E	PUSHL	#1		
			EF	9F	00170	PUSHAB	FDL_DEST		
00000000G	EF		07	FB	00176	CALLS	#7,PASS\$OPEN2		
			EF	9F	0017D	PUSHAB	FDL_DEST		: 0235
			01	FB	00183	CALLS	#1,PASS\$CLOSE2		
00000000G	EF		00	E1	0018A	BBC	#0,ISAM_ORG,11\$: 0242
00V00000000G	EF		00	FB	00192	CALLS	#0,SHUFFLE_AREAS		: 0244
00000000G	EF		03	DD	00199	PUSHL	#3		: 0249
		FEF1	CD	9F	0019B	PUSHAB	TEMP_STRING255		
		000000FF	8F	DD	0019F	PUSHL	#255		

		01	DD	001A5	PUSHL	#1		
		EF	9F	001A7	PUSHAB	FDL_DEST		
00000000G	EF	05	FB	001AD	CALLS	#5,PASSOPEN2		
		EF	9F	001B4	PUSHAB	FDL_DEST		: 0250
00000000G	EF	01	FB	001BA	CALLS	#1,PASSREWRITE2		
00000000G	EF	00	FB	001C1	CALLS	#0,GENERATE_FDL		: 0255
		EF	9F	001C8	PUSHAB	FDL_DEST		: 0260
00000000G	EF	01	FB	001CE	CALLS	#1,PASSCLOSE2		
00000000G	EF	01	90	001D5	MOVB	#1,FILE_CREATED		: 0265
		04	001DC	13\$:	RET			: 0275

; Routine Size: 477 bytes, Routine Base: \$CODE + 003F0

				00000	ADD_FDL_LINE:			: 0323
			003C	00000	.WORD	^M<R2,R3,R4,R5>		
		SE	80	AE	9E	00002	MOVAB	-128(SP),SP
	D1	AD	00000000G	EF	7D	00006	MOVQ	NULL_STRING,SAVE+17
	00000011G	EF	00000000G	EF	7D	0000E	MOVQ	NULL_STRING,TEST+17
	00000000G	EF		01	90	00019	MOVB	#1,FOLL_CHOICE
			00000047	8F	DF	00020	PUSHAL	#71
	00000000G	EF		01	FB	00026	CALLS	#1,QUERY
	00000000G	EF		01	90	0002D	MOVB	#1,FULL_CHOICE
	00000000G	EF		00	FB	00034	CALLS	#0,ASK_TEST_SECONDARY
CO	AD	00000000G	EF	0040	8F	28	0003B	MOVC3
			000000FC	8F	DD	00046	PUSHL	#64,TEST,SAVE
				07	DD	0004C	PUSHL	#252
				04	DD	0004E	PUSHL	#7
			00000000G	EF	9F	00050	PUSHAB	#4
				0B	DD	00056	PUSHL	SYSSOUTPUT_NAME
				01	DD	00058	PUSHL	#11
			00000000G	EF	9F	0005A	PUSHAB	#1
	00000000G	EF		07	FB	00060	PUSHAB	FDL_DEST
	00000000G	EF		EF	9F	00067	CALLS	#7,PASSOPEN2
	00000000G	EF		01	FB	0006D	PUSHAB	FDL_DEST
			00000040	8F	DD	00074	CALLS	#1,PASSREWRITE2
	00000000G	EF		01	FB	0007A	PUSHL	#64
	SC			50	DD	00081	CALLS	#1,PASSNEW2
6C	00000000G	EF	0040	8F	28	00084	MOVL	R0,DEF_TEST
			01	AC	D4	0008E	MOVC3	#64,TEST,(DEF_TEST)
			05	AC	D4	00091	CLRL	1(DEF_TEST)
			00000000G	EF	D0	00094	CLRL	5(DEF_TEST)
	52			EF	D0	00094	MOVL	DEF_CURRENT,SAVE_CURRENT
	00000000G	EF		5C	D0	0009B	MOVL	DEF_TEST,DEF_CURRENT
			00	8F	9F	000A2	PUSHAB	#0
	00000000G	EF		01	FB	000A5	CALLS	#1,SHOW_CUR_PRI_SEC
	00000000G	EF		52	DD	000AC	MOVL	SAVE_CURRENT,DEF_CURRENT
				5C	DD	000B3	PUSHL	DEF_TEST
	00000000G	EF		01	FB	000B5	CALLS	#1,PASSDISPOSE2
			00000000G	EF	9F	000BC	PUSHAB	FDL_DEST
	00000000G	EF		01	FB	000C2	CALLS	#1,PASSCLOSE2
			DF	AD	9F	000C9	PUSHAB	SAVE+31
			DE	AD	9F	000CC	PUSHAB	SAVE+30
			DA	AD	9F	000CF	PUSHAB	SAVE+26
			D9	AD	9F	000D2	PUSHAB	SAVE+25
			CO	AD	9F	000D5	PUSHAB	SAVE
	00000000G	EF		05	FB	000D8	CALLS	#5,FIND_OBJECT
	00V			50	E9	000DF	BLBC	EXISTS,3\$
			00000019	8F	DF	000E2	PUSHAL	#25

Generated Code			
00000000G	EF	01	FB 000E8
		00V	11 000EF
	50	01	90 000F1 3\$:
	03	50	E8 000F4 4\$:
		0000V	31 000F7
00000000G	EF	AD 0040	8F 28 000FA
		00000000G	EF 00 FB 00105
		00000000G	EF 00 FB 0010C
		80	AD 00000000G
80	BD	00000000G	EF 0040 8F 28 0011B
		00000000G	EF 00 8F 28 00126
		0F	19 A0 91 0012D
			00V 12 00131
	50	00000000G	EF 00 8F 28 00133
			60 94 0013A
			00V 11 0013C
	50	00000000G	EF 00 8F 28 0013E 7\$:
	60		01 90 00145
		00000000	8F DF 00148 8\$:
00000000G	EF	01	FB 0014E
	0F	00000019G	EF 91 00155
			03 12 0015C
		0000V	31 0015E
00000000G	EF	00000000G	EF 00 8F 28 00161
			5C 94 0016C
	50	00000000G	EF 00 8F 28 0016E 10\$:
	DA	AD 1A	A0 D1 00175
			00V 12 0017A
	50	00000000G	EF 00 8F 28 0017C
			60 95 00183
			00V 12 00185
	50	00000000G	EF 00 8F 28 00187
	D9	AD 19	A0 91 0018E
			00V 12 00193
	5C		01 90 00195
			00V 11 00198
00000000G	EF	00	FB 0019A 14\$:
	00V	5C	E8 001A1 15\$:
		00000000G	EF 00 8F 28 001A4
			C2 12 001AA
	00V	5C	E8 001AC 17\$:
		00000000G	EF 94 001AF
00000019G	EF	D9	AD 90 001B5
0000001AG	EF	DA	AD 00 8F 28 001BD
		0000001EG	EF 94 001C5
		0000001FG	EF 04 001CB
00000011G	EF	00000000G	EF 7D 001D1
00000009G	EF	00000000G	EF 7D 001DC
00000000G	EF	00	FB 001E7
80	BD	00000000G	EF 00 8F 28 001EE
		00000000	8F DF 00201
00000000G	EF	000000003	01 FB 00207 20\$:
		00000000G	EF 01 8F DF 0020E
			04 DD 0021B
			04 DD 00221
			CALLS #1, QUERY
			BRB 4\$
			MOV B #1, PROCEED : 0375
			BLBS PROCEED, .+3 : 0377
			BRW 21\$
			MOV C3 #64, SAVE, TEST : 0381
			CALLS #0, ASK TEST SECONDARY_VALUE : 0383
			CALLS #0, MAKE SCRATCH : 0385
			MOVL DEF_SCRATCH, -128(FP) : 0387
			MOV C3 #64, TEST, a-128(FP)
			MOVL DEF_SCRATCH, R0 : 0389
			CMPB 25(R0), #15
			BNEQ 7\$
			MOVL DEF_SCRATCH, R0 : 0391
			CLRB (R0)
			BRB 8\$
			MOVL DEF_SCRATCH, R0 : 0395
			MOV B #1, (R0)
			PUSHAL #0 : 0399
			CALLS #1, INSERT_IN_ORDER
			CMPB TEST+25, #15 : 0401
			BNEQ .+3
			BRW 20\$
			MOVL DEF_HEAD, DEF_CURRENT : 0408
			CLRB FOUND_PRI : 0409
			MOVL DEF_CURRENT, R0 : 0413
			CMPB 26(R0), SAVE+26
			BNEQ 14\$
			MOVL DEF_CURRENT, R0
			TSTB (R0)
			BNEQ 14\$
			MOVL DEF_CURRENT, R0
			CMPB 25(R0), SAVE+25
			BNEQ 14\$
			MOV B #1, FOUND_PRI : 0421
			BRB 15\$
			CALLS #0, INCR CURRENT : 0425
			BLBS FOUND_PRI, 17\$
			TSTL DEF_CURRENT
			BNEQ 10\$
			BLBS FOUND_PRI, 20\$: 0429
			CLRB TEST : 0433
			MOV B SAVE+25, TEST+25 : 0434
			MOVL SAVE+26, TEST+26 : 0435
			CLRB TEST+30 : 0436
			CLRL TEST+31 : 0437
			MOVQ NULL_STRING, TEST+17 : 0438
			MOVQ NULL_STRING, TEST+9 : 0439
			CALLS #0, MAKE SCRATCH : 0441
			MOVL DEF_SCRATCH, -128(FP) : 0443
			MOV C3 #64, TEST, a-128(FP)
			PUSHAL #0 : 0445
			CALLS #1, INSERT_IN_ORDER
			PUSHAL #3 : 0451
			CALLS #1, CLEAR
			PUSHAB SHIFT : 0453
			PUSHL #4

```
Generated Code

00000000G EF 00000000G EF 9F 00223 PUSHAB PASSFV OUTPUT
00000000G EF 03 FB 00229 CALLS #3,PASSWRITE_STRING
00000000G 7E 00000000G EF 01 DD 00230 PUSHL #1
00000000G EF 9A 00232 MOVZBL TAB, -(SP)
00000000G EF 9F 00239 PUSHAB PASSFV OUTPUT
00000000G EF 03 FB 0023F CALLS #3,PASSWRITE_CHAR
00000000G 7E 00000000G EF 01 DD 00246 PUSHL #1
00000000G EF 9A 00248 MOVZBL TAB, -(SP)
00000000G EF 9F 0024F PUSHAB PASSFV OUTPUT
00000000G EF 03 FB 00255 CALLS #3,PASSWRITE_CHAR
00000000G EF 9F 0025C PUSHAB ANSI_REVERSE
00000000G EF 04 DD 00262 PUSHL #4
00000000G EF 9F 00264 PUSHAB PASSFV OUTPUT
00000000G EF 03 FB 0026A CALLS #3,PASSWRITE_STRING
00000000G EF FFFF7F0 EF 9F 00271 PUSHAB C.AAB
00000000G 1B DD 00277 PUSHL #27
00000000G EF 9F 00279 PUSHAB PASSFV OUTPUT
00000000G EF 03 FB 0027F CALLS #3,PASSWRITE_STRING
00000000G EF 9F 00286 PUSHAB ANSI_RESET
00000000G EF 04 DD 0028C PUSHL #4
00000000G EF 9F 0028E PUSHAB PASSFV OUTPUT
00000000G EF 03 FB 00294 CALLS #3,PASSWRITE_STRING
00000000G EF 9F 0029B PUSHAB CRLF
00000000G EF 02 DD 002A1 PUSHL #2
00000000G EF 9F 002A3 PUSHAB PASSFV OUTPUT
00000000G EF 03 FB 002A9 CALLS #3,PASSWRITE_STRING
00000000G EF 9F 002B0 PUSHAB PASSFV OUTPUT
00000000G EF 01 FB 002B6 CALLS #1,PASSWRITELN2
000000FC 8F DD 002BD PUSHL #252 ; 0457
00000000G 07 DD 002C3 PUSHL #7
00000000G 04 DD 002C5 PUSHL #4
00000000G EF 9F 002C7 PUSHAB SYSSOUTPUT_NAME
00000000G 0B DD 002CD PUSHL #11
00000000G 01 DD 002CF PUSHL #1
00000000G EF 9F 002D1 PUSHAB FDL_DEST
00000000G EF 07 FB 002D7 CALLS #7,PASSOPEN2
00000000G EF 9F 002DE PUSHAB FDL_DEST ; 0458
00000000G EF 01 FB 002E4 CALLS #1,PASSREWRITE2
00000000G EF C0 AD 9F 002EB PUSHAB SAVE ; 0460
00000000G EF 01 FB 002EE CALLS #1,SHOW_PRIMARY_SECTION
00000000G EF 9F 002F5 PUSHAB FDL_DEST ; 0462
00000000G EF 01 FB 002FB CALLS #1,PASSCLOSE2
00000000G EF C0 AD 0040 8F 28 00302 21$: MOV C3 #64,SAVE,TEST ; 0466
00000000G EF 0000001F 8F DF 0030D PUSHL #31 ; 0468
00000000G EF 01 FB 00313 CALLS #1,QUERY ; 0470
00000000G 04 0031A RET

; Routine Size: 795 bytes, Routine Base: $CODE + 005CD

00000000G EF 00000000G EF 0000 00000 CHECK_DEFAULT: ; 0514
00000000G EF 03 D1 00002 .WORD ^M<> ; 0521
00000000G EF 12 0000D CMPL DEF_HEAD,DEF_TAIL
00000000G EF 31 0000F BNEQ +3
00000000G EF D0 00012 BRW i2$
00000000G EF 94 0001D MOV L DEF_HEAD,DEF_CURRENT ; 0528
00000019G EF 90 00023 CLRB TEST ; 0529
MOV B DEFAULT_PRIMARY,TEST+25 ; 0530
```

Generated Code			
0000001AG	EF	00000000G	EF D0 0002E
			5C 94 00039
		00	8F 9F 0003B 2\$:
		00000000G	EF 9F 0003E
00000000G	EF	00000000G	02 FB 00044
	00V		50 E9 0004B
	5C		01 90 0004E
			00V 11 00051
00000000G	EF	00000000G	00 FB 00053 4\$:
			EF D5 0005A 5\$:
			00V 13 00060
	D6		5C E9 00062
	00V		5C E8 00065 7\$:
00000000G	EF	00000000G	EF D0 00068
	5C	00000000G	EF D0 00073
	09	19	AC 91 0007A
			00V 12 0007E
00000000G	EF	00000000G	00 FB 00080
	50	00000000G	EF D0 00087 10\$:
00000000G	EF	19	A0 90 0008E
	50	00000000G	EF D0 00096
00000000G	EF	1A	A0 D0 0009D
00000000G	EF	00000000G	EF D0 000A5
			04 000B0 12\$:
			RETI
			MOVL DEFAULT_PRINUM,TEST+26 : 0531
			CLRB FOUND_PRIMARY : 0532
			PUSHAB #0 : 0536
			PUSHAB TEST
			CALLS #2,CURRENT_EQ_TEST
			BLBC R0,4\$
			MOVB #1,FOUND_PRIMARY : 0538
			BRB 5\$
			CALLS #0,INCR CURRENT : 0542
			TSTL DEF_CURRENT
			BEQL 7\$
			BLBC FOUND_PRIMARY,2\$
			BLBS FOUND_PRIMARY,12\$: 0546
			MOVL DEF_HEAD,DEF_CURRENT : 0553
			MOVL DEF_CURRENT,R12 : 0555
			CMPB 25(R12),#9
			BNEQ 10\$
			CALLS #0,INCR CURRENT : 0557
			MOVL DEF_CURRENT,R0 : 0562
			MOVB 25(R0),DEFAULT_PRIMARY
			MOVL DEF_CURRENT,R0 : 0563
			MOVL 26(R0),DEFAULT_PRINUM
			MOVL DEFAULT_PRINUM,INPUT_NUMBER : 0564
			RETI : 0570

; Routine Size: 177 bytes, Routine Base: \$CODE + 008E8

			00000	DELETE_FDL_LINE:		: 0627
			003C 00000	.WORD	^M<R2,R3,R4,R5>	
			9E 00002	MOVAB	-64(SP),SP	
00000000G	5E	CO	AE 9E 00002	CMPL	DEF_HEAD,DEF_TAIL	: 0642
	EF	00000000G	EF D1 00006	BNEQ	+3	
			03 12 00011	BRW	36\$	
			0000V 31 00013	MOVQ	NULL_STRING,SAVE+17	: 0646
	D1	AD	EF 7D 00016	MOVQ	NULL_STRING,TEST+17	: 0647
00000011G	EF	00000000G	EF 7D 0001E	CALLS	#0,CHECK_DEFAULT	: 0649
08E8	CF		00 FB 00029	CLRB	FULL_CHOICE	: 0655
			EF 94 0002E	PUSHAL	#71	: 0656
			8F DF 00034	CALLS	#1,QUERY	
00000000G	EF		01 FB 0003A	CLRB	NO_MORE_PRI	: 0658
			5C 94 00041	CLRB	FULL_CHOICE	: 0660
			EF 94 00043	CALLS	#0,ASK_TEST_SECONDARY	: 0661
00000000G	EF	00000000G	00 FB 00049	MOVCB	#64,TEST,SAVE	: 0666
			8F 28 00050	PUSHAB	TEST+31	: 0668
			EF 9F 0005B	PUSHAB	TEST+30	
			EF 9F 00061	PUSHAB	TEST+26	
			EF 9F 00067	PUSHAB	TEST+25	
			EF 9F 0006D	PUSHAB	TEST	
			EF 9F 00073	CALLS	#5,FIND_OBJECT	
00000000G	EF		05 FB 00079	PUSHL	#252	: 0676
			8F DD 00080	PUSHL	#7	
			07 DD 00086	PUSHL	#4	
			04 DD 00088	PUSHAB	SYSS\$OUTPUT_NAME	
			EF 9F 0008A	PUSHL	#11	
			0B DD 00090	PUSHL	#1	
			01 DD 00092	PUSHAB	FDL_DEST	
			EF 9F 00094	CALLS	#7,PASS\$OPEN2	
00000000G	EF	00000000G	07 FB 0009A			

Generated Code								
00000000G	EF	00000000G	EF	9F	000A1	PUSHAB	FDL_DEST	: 0677
			01	FB	000A7	CALLS	#1,PASSREWRITE2	
00000000G	EF	01	8F	9F	000AE	PUSHAB	#1	: 0679
00000000G	EF	00000000G	01	FB	000B1	CALLS	#1,SHOW_CUR_PRI_SEC	
00000000G	EF	00000000G	EF	9F	000B8	PUSHAB	FDL_DEST	: 0681
00000000G	EF	0000001F	01	FB	000BE	CALLS	#1,PASSCLOSE2	
00000000G	EF		8F	DF	000C5	PUSHAL	#31	: 0683
00000000G	EF		01	FB	000CB	CALLS	#1,QUERY	
00000000G	EF		00	FB	000D2	CALLS	#0,DELETE_CURRENT	: 0685
	OF	00000019G	EF	91	000D9	CMPB	TEST+25,#T5	: 0687
			03	12	000E0	BNEQ	+3	
		0000V	31	000E2	BRW	35\$		
			52	94	000E5	CLRB	REMAINING_PRI	: 0694
			53	94	000E7	CLRB	REMAINING_SEC	: 0695
00000000G	EF	00000000G	EF	D0	000E9	MOVL	DEF_HEAD,DEF_CURRENT	: 0697
	50	00000000G	EF	D0	000F4	MOVL	DEF_CURRENT,R0	: 0701
DA	AD	1A	A0	D1	000FB	CMPL	26(R0),SAVE+26	
			00V	12	00100	BNEQ	13\$	
	50	00000000G	EF	D0	00102	MOVL	DEF_CURRENT,R0	
D9	AD	19	A0	91	00109	CMPB	25(R0),SAVE+25	
			00V	12	0010E	BNEQ	13\$	
	50	00000000G	EF	D0	00110	MOVL	DEF_CURRENT,R0	: 0709
			60	95	00117	TSTB	(R0)	
			00V	12	00119	BNEQ	9\$	
	52		01	90	0011B	MOVB	#1,REMAINING_PRI	: 0713
	54	00000000G	EF	D0	0011E	MOVL	DEF_CURRENT,DEF_REM_PRI	: 0714
			00V	11	00125	BRB	13\$	
	50	00000000G	EF	D0	00127	MOVL	DEF_CURRENT,R0	: 0718
	01		60	91	0012E	CMPB	(R0),#1	
			00V	12	00131	BNEQ	13\$	
	53		01	90	00133	MOVB	#1,REMAINING_SEC	: 0720
00000000G	EF		00	FB	00136	CALLS	#0,INCR_CURRENT	: 0724
	00V		52	E9	0013D	BLBC	REMAINING_PRI,15\$	
	00V		53	E8	00140	BLBS	REMAINING_SEC,16\$	
		00000000G	EF	D5	00143	TSTL	DEF_CURRENT	
			A9	12	00149	BNEQ	5\$	
	03		52	E8	0014B	BLBS	REMAINING_PRI,..+3	: 0728
		0000V	31	0014E	BRW	19\$		
	03		53	E9	00151	BLBC	REMAINING_SEC,..+3	
		0000V	31	00154	BRW	19\$		
		00000000G	EF	9F	00157	PUSHAB	CRLF	: 0736
			02	DD	0015D	PUSHL	#2	
		00000000G	EF	9F	0015F	PUSHAB	PASSFV_OUTPUT	
00000000G	EF		03	FB	00165	CALLS	#3,PASSWRITE_STRING	
		00000000G	EF	9F	0016C	PUSHAB	SHIFT	
			04	DD	00172	PUSHL	#4	
		00000000G	EF	9F	00174	PUSHAB	PASSFV_OUTPUT	
00000000G	EF		03	FB	0017A	CALLS	#3,PASSWRITE_STRING	
		00000000G	EF	9F	00181	PUSHAB	ANSI_REVERSE	
			04	DD	00187	PUSHL	#4	
		00000000G	EF	9F	00189	PUSHAB	PASSFV_OUTPUT	
00000000G	EF		03	FB	0018F	CALLS	#3,PASSWRITE_STRING	
		FFFFF51B	EF	9F	00196	PUSHAB	C.AAC	
			3A	DD	0019C	PUSHL	#58	
		00000000G	EF	9F	0019E	PUSHAB	PASSFV_OUTPUT	
00000000G	EF		03	FB	001A4	CALLS	#3,PASSWRITE_STRING	
		00000000G	EF	9F	001AB	PUSHAB	ANSI_RESET	

00000000G	EF	00000000G	04	DD	001B1	PUSHL	#4		
			EF	9F	001B3	PUSHAB	PASSFV_OUTPUT		
			03	FB	001B9	CALLS	#3,PASSWRITE_STRING		
00000000G	EF	00000000G	EF	9F	001C0	PUSHAB	PASSFV_OUTPUT		
00000000G	EF		01	FB	001C6	CALLS	#1,PASSWriteln2		
00000000G	EF		54	DO	001CD	MOVL	DEF_REM_PRI,DEF_CURRENT		: 0740
00000000G	EF		00	FB	001D4	CALLS	#0,DELETE_CURRENT		: 0741
	5C		01	90	001DB	MOVB	#1,NO_MORE_PRI		: 0742
		00004140	8F	DF	001DE	PUSHAF	#^F3.0		: 0744
00000000G	EF		01	FB	001E4	CALLS	#1,LIB\$WAIT		
		0000V	31	001EB	BRW	31\$			
	00V		52	E8	001EE	19\$:	BLBS	REMAINING_PRI,22\$: 0748
	03		53	E9	001F1	BLBC	REMAINING_SEC,..+3		
		0000V	31	001F4	BRW	31\$			
	03		52	E9	001F7	22\$:	BLBC	REMAINING_PRI,..+3	: 0756
		0000V	31	001FA	BRW	25\$			
	03		53	E9	001FD	BLBC	REMAINING_SEC,..+3		
		0000V	31	00200	BRW	25\$			
		00000000G	EF	9F	00203	PUSHAB	CRLF		: 0764
			02	DD	00209	PUSHL	#2		
00000000G	EF	00000000G	EF	9F	0020B	PUSHAB	PASSFV_OUTPUT		
			03	FB	00211	CALLS	#3,PASSWRITE_STRING		
		00000000G	EF	9F	00218	PUSHAB	SHIFT		
			04	DD	0021E	PUSHL	#4		
00000000G	EF	00000000G	EF	9F	00220	PUSHAB	PASSFV_OUTPUT		
			03	FB	00226	CALLS	#3,PASSWRITE_STRING		
		00000000G	EF	9F	0022D	PUSHAB	ANSI_REVERSE		
			04	DD	00233	PUSHL	#4		
00000000G	EF	00000000G	EF	9F	00235	PUSHAB	PASSFV_OUTPUT		
			03	FB	0023B	CALLS	#3,PASSWRITE_STRING		
		FFFFF4AB	EF	9F	00242	PUSHAB	C.AAD		
			35	DD	00248	PUSHL	#53		
00000000G	EF	00000000G	EF	9F	0024A	PUSHAB	PASSFV_OUTPUT		
			03	FB	00250	CALLS	#3,PASSWRITE_STRING		
		00000000G	EF	9F	00257	PUSHAB	ANSI_RESET		
			04	DD	0025D	PUSHL	#4		
00000000G	EF	00000000G	EF	9F	0025F	PUSHAB	PASSFV_OUTPUT		
			03	FB	00265	CALLS	#3,PASSWRITE_STRING		
00000000G	EF	00000000G	EF	9F	0026C	PUSHAB	PASSFV_OUTPUT		
			01	FB	00272	CALLS	#1,PASSWriteln2		
	5C		01	90	00279	MOVB	#1,NO_MORE_PRI		: 0767
		00004100	8F	DF	0027C	PUSHAF	#^F2.0		: 0769
00000000G	EF		01	FB	00282	CALLS	#1,LIB\$WAIT		
		0000V	31	00289	BRW	31\$			
	03		52	E8	0028C	25\$:	BLBS	REMAINING_PRI,..+3	: 0773
		0000V	31	0028F	BRW	31\$			
	03		53	E8	00292	BLBS	REMAINING_SEC,..+3		
		0000V	31	00295	BRW	31\$			
		00000003	8F	DF	00298	PUSHAL	#3		: 0781
00000000G	EF		01	FB	0029E	CALLS	#1,CLEAR		
		00000000G	EF	9F	002A5	PUSHAB	SHIFT		: 0783
			04	DD	002AB	PUSHL	#4		
		00000000G	EF	9F	002AD	PUSHAB	PASSFV_OUTPUT		
00000000G	EF		03	FB	002B3	CALLS	#3,PASSWRITE_STRING		
			01	DD	002BA	PUSHL	#1		
	7E	00000000G	EF	9A	002BC	MOVZBL	TAB,-(SP)		
		00000000G	EF	9F	002C3	PUSHAB	PASSFV_OUTPUT		

Generated Code						
00000000G	EF	03	FB	002C9	CALLS	#3,PASSWRITE_CHAR
		01	DD	002D0	PUSHL	#1
	7E	00000000G	EF	9A	002D2	MOVZBL TAB, -(SP)
		00000000G	EF	9F	002D9	PUSHAB PASSFV_OUTPUT
00000000G	EF	03	FB	002DF	CALLS	#3,PASSWRITE_CHAR
		00000000G	EF	9F	002E6	PUSHAB ANSI_REVERSE
			04	DD	002EC	PUSHL #4
		00000000G	EF	9F	002EE	PUSHAB PASSFV_OUTPUT
00000000G	EF	03	FB	002F4	CALLS	#3,PASSWRITE_STRING
		FFFFF42A	EF	9F	002FB	PUSHAB C.AAE
			1B	DD	00301	PUSHL #27
		00000000G	EF	9F	00303	PUSHAB PASSFV_OUTPUT
00000000G	EF	03	FB	00309	CALLS	#3,PASSWRITE_STRING
		00000000G	EF	9F	00310	PUSHAB ANSI_RESET
			04	DD	00316	PUSHL #4
		00000000G	EF	9F	00318	PUSHAB PASSFV_OUTPUT
00000000G	EF	03	FB	0031E	CALLS	#3,PASSWRITE_STRING
		00000000G	EF	9F	00325	PUSHAB CRLF
			02	DD	0032B	PUSHL #2
		00000000G	EF	9F	0032D	PUSHAB PASSFV_OUTPUT
00000000G	EF	03	FB	00333	CALLS	#3,PASSWRITE_STRING
		00000000G	EF	9F	0033A	PUSHAB PASSFV_OUTPUT
00000000G	EF	01	FB	00340	CALLS	#1,PASSWRITELN2
		000000FC	8F	DD	00347	PUSHL #252
			07	DD	0034D	PUSHL #7
			04	DD	0034F	PUSHL #4
		00000000G	EF	9F	00351	PUSHAB SYS\$OUTPUT_NAME
			0B	DD	00357	PUSHL #11
			01	DD	00359	PUSHL #1
		00000000G	EF	9F	0035B	PUSHAB FDL_DEST
00000000G	EF	07	FB	00361	CALLS	#7,PASSOPEN2
		00000000G	EF	9F	00368	PUSHAB FDL_DEST
00000000G	EF	01	FB	0036E	CALLS	#1,PASSREWRITE2
		C0	AD	9F	00375	PUSHAB SAVE
00000000G	EF	01	FB	00378	CALLS	#1,SHOW_PRIMARY_SECTION
		00000000G	EF	9F	0037F	PUSHAB FDL_DEST
00000000G	EF	01	FB	00385	CALLS	#1,PASSCLOSE2
00000019G	EF	D9	AD	90	0038C	31\$: MOVB SAVE+25,TEST+25
0000001AG	EF	DA	AD	D0	00394	31\$: MOVL SAVE+26,TEST+26
	00V		5C	E8	0039C	31\$: BLBS NO_MORE_PRI,35\$
		0000001F	8F	DF	0039F	31\$: PUSHAL #3T
00000000G	EF	01	FB	003A5	CALLS	#1,QUERY
		00V	11	003AC	35\$: BRB 37\$	
		00000000G	EF	9F	003AE	36\$: PUSHAB SHIFT
			04	DD	003B4	PUSHL #4
		00000000G	EF	9F	003B6	PUSHAB PASSFV_OUTPUT
00000000G	EF	03	FB	003BC	CALLS	#3,PASSWRITE_STRING
		00000000G	EF	9F	003C3	PUSHAB ANSI_REVERSE
			04	DD	003C9	PUSHL #4
		00000000G	EF	9F	003CB	PUSHAB PASSFV_OUTPUT
00000000G	EF	03	FB	003D1	CALLS	#3,PASSWRITE_STRING
		FFFFF369	EF	9F	003D8	PUSHAB C.AAF
			22	DD	003DE	PUSHL #34
		00000000G	EF	9F	003E0	PUSHAB PASSFV_OUTPUT
00000000G	EF	03	FB	003E6	CALLS	#3,PASSWRITE_STRING
		00000000G	EF	9F	003ED	PUSHAB ANSI_RESET
			04	DD	003F3	PUSHL #4

00000000G	EF	00000000G	EF	9F	003F5	PUSHAB	PASS\$FV OUTPUT	
			03	FB	003FB	CALLS	#3,PASS\$WRITE STRING	
00000000G	EF	00000000G	EF	9F	00402	PUSHAB	PASS\$FV OUTPUT	
			01	FB	00408	CALLS	#1,PASS\$WRITELN2	
00000000G	EF	00004140	8F	DF	0040F	PUSHAF	#^f3.0	; 0815
			01	FB	00415	CALLS	#1,LIB\$WAIT	
			04	0041C	37\$:	RET		; 0819

; Routine Size: 1053 bytes, Routine Base: \$CODE + 00999

				00000	MODIFY_FDL_LINE:			; 0868	
				003C	00000	.WORD	^M<R2,R3,R4,R5>		
				9E	00002	MOVAB	-128(SP),SP		
00000000G	SE	80	AE	D1	00006	CMPL	DEF_HEAD,DEF_TAIL	; 0879	
	EF	00000000G	EF	12	00011	BNEQ	+3		
			0000V	31	00013	BRW	7\$		
	D1	AD	00000000G	EF	7D	00016	MOVQ	NULL_STRING,SAVE+17	; 0883
	C9	AD	00000000G	EF	7D	0001E	MOVQ	NULL_STRING,SAVE+9	; 0884
00000011G	EF	00000000G	EF	7D	00026	MOVQ	NULL_STRING,TEST+17	; 0885	
00000009G	EF	00000000G	EF	7D	00031	MOVQ	NULL_STRING,TEST+9	; 0886	
08E8	CF		00	FB	0003C	CALLS	#0,CHECK_DEFAULT	; 0888	
		00000000G	EF	94	00041	CLRB	FULL_CHOICE	; 0894	
		000000047	8F	DF	00047	PUSHAL	#71	; 0895	
00000000G	EF		01	FB	0004D	CALLS	#1,QUERY		
		00000000G	EF	94	00054	CLRB	FULL_CHOICE	; 0897	
00000000G	EF		00	FB	0005A	CALLS	#0,ASK_TEST_SECONDARY	; 0898	
		0000001FG	EF	9F	00061	PUSHAB	TEST+3T	; 0900	
		0000001EG	EF	9F	00067	PUSHAB	TEST+30		
		0000001AG	EF	9F	0006D	PUSHAB	TEST+26		
		00000019G	EF	9F	00073	PUSHAB	TEST+25		
		00000000G	EF	9F	00079	PUSHAB	TEST		
00000000G	EF		05	FB	0007F	CALLS	#5,FIND_OBJECT		
	50	00000000G	EF	D0	00086	MOVL	DEF_CURRENT,RO	; 0905	
CO	AD	0040	8F	28	0008D	MOVC3	#64,(RO),SAVE		
		000000FC	8F	DD	00094	PUSHL	#252	; 0910	
			07	DD	0009A	PUSHL	#7		
			04	DD	0009C	PUSHL	#4		
		00000000G	EF	9F	0009E	PUSHAB	SYSS\$OUTPUT_NAME		
			0B	DD	000A4	PUSHL	#11		
			01	DD	000A6	PUSHL	#1		
		00000000G	EF	9F	000A8	PUSHAB	FDL_DEST		
00000000G	EF		07	FB	000AE	CALLS	#7,PASS\$OPEN2	; 0911	
		00000000G	EF	9F	000B5	PUSHAB	FDL_DEST		
00000000G	EF		01	FB	000BB	CALLS	#1,PASS\$REWRITE2	; 0913	
		01	8F	9F	000C2	PUSHAB	#1		
00000000G	EF		01	FB	000C5	CALLS	#1,SHOW_CUR_PRI_SEC	; 0915	
		00000000G	EF	9F	000CC	PUSHAB	FDL_DEST		
00000000G	EF		01	FB	000D2	CALLS	#1,PASS\$CLOSE2	; 0917	
	CO	AD	0040	8F	28	000D9	MOVC3	#64,SAVE_TEST	; 0919
00000000G	EF		00	FB	000E4	CALLS	#0,ASK_TEST_SECONDARY_VALUE	; 0921	
00000000G	EF		00	FB	000EB	CALLS	#0,MAKE_SCRATCH	; 0923	
	80	BD	00000000G	EF	D0	000F2	MOVL	DEF_SCRATCH,-128(FP)	
			EF	8F	28	000FA	MOVC3	#64,TEST,@-128(FP)	
		00000000G	EF	D0	00105	MOVL	DEF_SCRATCH,RO	; 0925	
		19	AQ	91	0010C	CMPB	25(RO),#15		
			00V	12	00110	BNEQ	4\$		
		50	00000000G	EF	D0	00112	MOVL	DEF_SCRATCH,RO	; 0927

		60	94	00119	CLRB	(R0)	
		00V	11	0011B	BRB	5\$	
50	00000000G	EF	D0	0011D	4\$:	MOVL	DEF_SCRATCH,R0 ; 0931
60	00000000	01	90	00124	MOVB	#1,(R0)	
00000000G	EF	8F	DF	00127	5\$:	PUSHAL	#0 ; 0933
00000000G	EF	01	FB	0012D	CALLS	#1,INSERT_IN_ORDER	
00000000G	EF	8F	DF	00134	PUSHAL	#3	; 0935
00000000G	EF	01	FB	0013A	CALLS	#1,CLEAR	
00000000G	EF	04	9F	00141	PUSHAB	SHIFT	; 0937
00000000G	EF	04	DD	00147	PUSHL	#4	
00000000G	EF	03	9F	00149	PUSHAB	PASS\$FV_OUTPUT	
00000000G	EF	01	FB	0014F	CALLS	#3,PASS\$WRITE_STRING	
00000000G	EF	01	DD	00156	PUSHL	#1	
00000000G	EF	03	9A	00158	MOVZBL	TAB,-(SP)	
00000000G	EF	03	9F	0015F	PUSHAB	PASS\$FV_OUTPUT	
00000000G	EF	01	FB	00165	CALLS	#3,PASS\$WRITE_CHAR	
00000000G	EF	01	DD	0016C	PUSHL	#1	
00000000G	EF	03	9A	0016E	MOVZBL	TAB,-(SP)	
00000000G	EF	03	9F	00175	PUSHAB	PASS\$FV_OUTPUT	
00000000G	EF	04	FB	0017B	CALLS	#3,PASS\$WRITE_CHAR	
00000000G	EF	04	9F	00182	PUSHAB	ANSI_REVERSE	
00000000G	EF	04	DD	00188	PUSHL	#4	
00000000G	EF	03	9F	0018A	PUSHAB	PASS\$FV_OUTPUT	
00000000G	EF	03	FB	00190	CALLS	#3,PASS\$WRITE_STRING	
00000000G	EF	03	9F	00197	PUSHAB	C.AAG	
00000000G	EF	1B	DD	0019D	PUSHL	#27	
00000000G	EF	03	9F	0019F	PUSHAB	PASS\$FV_OUTPUT	
00000000G	EF	03	FB	001A5	CALLS	#3,PASS\$WRITE_STRING	
00000000G	EF	04	9F	001AC	PUSHAB	ANSI_RESET	
00000000G	EF	04	DD	001B2	PUSHL	#4	
00000000G	EF	03	9F	001B4	PUSHAB	PASS\$FV_OUTPUT	
00000000G	EF	03	FB	001BA	CALLS	#3,PASS\$WRITE_STRING	
00000000G	EF	02	9F	001C1	PUSHAB	CRLF	
00000000G	EF	02	DD	001C7	PUSHL	#2	
00000000G	EF	03	9F	001C9	PUSHAB	PASS\$FV_OUTPUT	
00000000G	EF	03	FB	001CF	CALLS	#3,PASS\$WRITE_STRING	
00000000G	EF	01	9F	001D6	PUSHAB	PASS\$FV_OUTPUT	
00000000G	EF	01	FB	001DC	CALLS	#1,PASS\$WRITELN2	
00000000G	EF	8F	DD	001E3	PUSHL	#252	; 0941
00000000G	EF	07	DD	001E9	PUSHL	#7	
00000000G	EF	04	DD	001EB	PUSHL	#4	
00000000G	EF	0B	9F	001ED	PUSHAB	SYSS\$OUTPUT_NAME	
00000000G	EF	01	DD	001F3	PUSHL	#11	
00000000G	EF	07	9F	001F7	PUSHL	#1	
00000000G	EF	07	FB	001FD	PUSHAB	FDL_DEST	
00000000G	EF	01	9F	00204	CALLS	#7,PASS\$OPEN2	; 0942
00000000G	EF	01	FB	0020A	PUSHAB	FDL_DEST	
00000000G	EF	AD	9F	00211	CALLS	#1,PASS\$REWRITE2	; 0944
00000000G	EF	01	FB	00214	PUSHAB	SAVE	
00000000G	EF	01	9F	0021B	CALLS	#1,SHOW_PRIMARY_SECTION	; 0946
00000000G	EF	01	FB	00221	PUSHAB	FDL_DEST	
00000000G	EF	8F	28	00228	CALLS	#1,PASS\$CLOSE2	; 0948
00000000G	EF	8F	DF	00233	MOVCL	#64,SAVE_TEST	; 0950
00000000G	EF	01	FB	00239	PUSHAL	#31	
00000000G	EF	00V	11	00240	CALLS	#1,QUERY	
00000000G	EF	9F	00242	7\$:	BRB	8\$	
					PUSHAB	SHIFT	; 0958

Generated Code

```
00000000G EF 00000000G 04 DD 00248      PUSHL #4
00000000G EF 00000000G EF 9F 0024A      PUSHAB PASS$V OUTPUT
00000000G EF 00000000G 03 FB 00250      CALLS #3,PASS$WRITE_STRING
00000000G EF 00000000G EF 9F 00257      PUSHAB ANSI_REVERSE
00000000G EF 00000000G 04 DD 0025D      PUSHL #4
00000000G EF 00000000G EF 9F 0025F      PUSHAB PASS$V OUTPUT
00000000G EF FFFFFFFF8 03 FB 00265      CALLS #3,PASS$WRITE_STRING
00000000G EF 00000000G EF 9F 0026C      PUSHAB C.AAH
00000000G EF 00000000G 22 DD 00272      PUSHL #34
00000000G EF 00000000G EF 9F 00274      PUSHAB PASS$V OUTPUT
00000000G EF 00000000G 03 FB 0027A      CALLS #3,PASS$WRITE_STRING
00000000G EF 00000000G EF 9F 00281      PUSHAB ANSI_RESET
00000000G EF 00000000G 04 DD 00287      PUSHL #4
00000000G EF 00000000G EF 9F 00289      PUSHAB PASS$V OUTPUT
00000000G EF 00000000G 03 FB 0028F      CALLS #3,PASS$WRITE_STRING
00000000G EF 00000000G EF 9F 00296      PUSHAB PASS$V OUTPUT
00000000G EF 00004140 01 FB 0029C      CALLS #1,PASS$WRITELN2
00000000G EF 00000000G 8F DF 002A3      PUSHAF #^F3.0
00000000G EF 00000000G 01 FB 002A9      CALLS #1,LIB$WAIT
04 002B0 8$:      RET
```

; 0961

; 0965

; Routine Size: 689 bytes, Routine Base: \$CODE + 00DB6

```
00000000G EF 00000000G 08 DD 00000      .WORD ^M<>
00000000G EF 00000000G 00 C2 00002      .WORD ^M<R2>
00000000G EF 00000000G 00 DD 00005      CLRB NON_EMPTY
00000000G EF 00000000G 00 DD 00008      CLRB ISAM_FDL
00000000G EF 00000000G 00 DD 0000D      CMPL DEF_READ,DEF_TAIL
00000000G EF 00000000G 00 DD 0000F      BNEQ 2$
00000000G EF 00000000G 00 DD 00011      MOVL DEF_HEAD,R0
00000000G EF 00000000G 00 DD 00013      CMPB 25(R0),#9
00000000G EF 00000000G 00 DD 0001A      BNEQ +3
00000000G EF 00000000G 00 DD 0001E      BRW 9$
00000000G EF 00000000G 00 DD 00020      MOVB #1,NON_EMPTY
00000000G EF 00000000G 04 FB 0004F      CALLS #4,LIB$SIGNAL
04 00056 2$:      RET
```

; 1012

; 1020

; 1032

; 1034

; 1036

; Routine Size: 87 bytes, Routine Base: \$CODE + 01067

```
00000000G EF 00000000G 5C 94 00002      CLRB NON_EMPTY
00000000G EF 00000000G 52 94 00004      CLRB ISAM_FDL
00000000G EF 00000000G 00V D1 00006      CMPL DEF_READ,DEF_TAIL
00000000G EF 00000000G 50 00000      BNEQ 2$
00000000G EF 00000000G 09 00000      MOVL DEF_HEAD,R0
00000000G EF 00000000G 03 00000      CMPB 25(R0),#9
00000000G EF 00000000G 00V 03 00001      BNEQ +3
00000000G EF 00000000G 5C 0000V 31 00020      BRW 9$
00000000G EF 00000000G 01 90 00023 2$:      MOVB #1,NON_EMPTY
```

; 1082

; 1090

; 1091

; 1096

; 1104

		00000000	8F	DF	00026	PUSHAL	#0		: 1110
		62	8F	9F	0002C	PUSHAB	#98		
		00000000	8F	DF	0002F	PUSHAL	#0		
		08	8F	9F	00035	PUSHAB	#8		
		01	8F	9F	00038	PUSHAB	#1		
00000000G	EF		05	FB	0003B	CALLS	#5,FIND_OBJECT		
	00V		50	E9	00042	BLBC	R0,6\$		
	50	00000000G	EF	D0	00045	MOVL	DEF,CURRENT,R0		: 1114
	1F	23	A0	D1	0004C	CMPL	35(R0),#31		
			00V	12	00050	BNEQ	6\$		
	52		01	90	00052	MOVB	#1,ISAM_FDL		: 1116
	03		52	E9	00055	BLBC	ISAM_FDL,..+3		: 1120
		0000V	31	00058	BRW	13\$			
		00000000G	EF	9F	0005B	PUSHAB	SHIFT		: 1124
			04	DD	00061	PUSHL	#4		
		00000000G	EF	9F	00063	PUSHAB	PASSFV_OUTPUT		
00000000G	EF		03	FB	00069	CALLS	#3,PASSWRITE_STRING		
		00000000G	EF	9F	00070	PUSHAB	ANSI_REVERSE		
			04	DD	00076	PUSHL	#4		
		00000000G	EF	9F	00078	PUSHAB	PASSFV_OUTPUT		
00000000G	EF		03	FB	0007E	CALLS	#3,PASSWRITE_STRING		
		FFFFE0FB	EF	9F	00085	PUSHAB	C.AAI		
			2F	DD	0008B	PUSHL	#47		
		00000000G	EF	9F	0008D	PUSHAB	PASSFV_OUTPUT		
00000000G	EF		03	FB	00093	CALLS	#3,PASSWRITE_STRING		
		00000000G	EF	9F	0009A	PUSHAB	ANSI_RESET		
			04	DD	000A0	PUSHL	#4		
		00000000G	EF	9F	000A2	PUSHAB	PASSFV_OUTPUT		
00000000G	EF		03	FB	000A8	CALLS	#3,PASSWRITE_STRING		
		00000000G	EF	9F	000AF	PUSHAB	PASSFV_OUTPUT		
00000000G	EF		01	FB	000B5	CALLS	#1,PASSWriteln2		
		00004140	8F	DF	000BC	PUSHAF	#AF3.0		: 1128
00000000G	EF		01	FB	000C2	CALLS	#1,LIB\$WAIT		
		0000V	31	000C9	BRW	13\$			
00V00000000G	EF		00	E0	000CC	BBS	#0,AUTO_TUNE,11\$: 1136
		00000000G	EF	9F	000D4	PUSHAB	SHIFT		: 1142
			04	DD	000DA	PUSHL	#4		
		00000000G	EF	9F	000DC	PUSHAB	PASSFV_OUTPUT		
00000000G	EF		03	FB	000E2	CALLS	#3,PASSWRITE_STRING		
		00000000G	EF	9F	000E9	PUSHAB	ANSI_REVERSE		
			04	DD	000EF	PUSHL	#4		
		00000000G	EF	9F	000F1	PUSHAB	PASSFV_OUTPUT		
00000000G	EF		03	FB	000F7	CALLS	#3,PASSWRITE_STRING		
		FFFFE0B2	EF	9F	000FE	PUSHAB	C.AAJ		
			26	DD	00104	PUSHL	#38		
		00000000G	EF	9F	00106	PUSHAB	PASSFV_OUTPUT		
00000000G	EF		03	FB	0010C	CALLS	#3,PASSWRITE_STRING		
		00000000G	EF	9F	00113	PUSHAB	ANSI_RESET		
			04	DD	00119	PUSHL	#4		
		00000000G	EF	9F	0011B	PUSHAB	PASSFV_OUTPUT		
00000000G	EF		03	FB	00121	CALLS	#3,PASSWRITE_STRING		
		00000000G	EF	9F	00128	PUSHAB	PASSFV_OUTPUT		
00000000G	EF		01	FB	0012E	CALLS	#1,PASSWriteln2		
		00004140	8F	DF	00135	PUSHAF	#AF3.0		: 1146
00000000G	EF		01	FB	0013B	CALLS	#1,LIB\$WAIT		
			00V	11	00142	BRB	13\$		
			00	DD	00144	PUSHL	#0		: 1151

```
00 DD 00146      PUSHL #0
00 DD 00148      PUSHL #0
00B3801C 8F DD 0014A  PUSHL #11763740
00000000G EF 04 FB 00150  CALLS #4,LIB$SIGNAL
52 52 92 00157 13$: MCOMB ISAM_FDL,R2 ; 1158
5C 52 8A 0015A  BICB2 R2,NON_EMPTY
50 5C 90 0015D  MOVB VERIFY_ISAM_DEFINITION,R0 ; 1160
04 00160      RET
```

; Routine Size: 353 bytes, Routine Base: \$CODE + 010BE

```
00000 REDESIGN_SCRIPT_PROC: ; 1206
0000 .WORD ^M<>
10BE CF 00 FB 00002  CALLS #0,VERIFY_ISAM_DEFINITION ; 1210
00V 50 E9 00007  BLBC R0,2$
00 8F 9F 0000A  PUSHAB #0 ; 1212
01 8F 9F 0000D  PUSHAB #1
00000000G EF 02 FB 00010  CALLS #2,INDEXED_DESIGN
04 00017 2$: RET ; 1214
```

; Routine Size: 24 bytes, Routine Base: \$CODE + 0121F

```
00000 ADD_KEY_SCRIPT_PROC: ; 1260
0000 .WORD ^M<>
10BE CF 00 FB 00002  CALLS #0,VERIFY_ISAM_DEFINITION ; 1264
00V 50 E9 00007  BLBC R0,4$
00 8F 9F 0000A  PUSHAB #0 ; 1271
00000000G EF 01 FB 0000D  CALLS #1,SCAN_DEFINITION
00000084G EF 00 D0 00014  MOVL HIGH_KEY, IDATA+132 ; 1276
00V00000000G EF 00 E1 0001F  BBC #0,FOUND_0,3$ ; 1278
00000084G EF 00 D6 00027  INCL IDATA+132 ; 1280
01 8F 9F 0002D 3$: PUSHAB #1 ; 1285
01 8F 9F 00030  PUSHAB #1
00000000G EF 02 FB 00033  CALLS #2,INDEXED_DESIGN
04 0003A 4$: RET ; 1289
```

; Routine Size: 59 bytes, Routine Base: \$CODE + 01237

```
00000 DELETE_KEY_SCRIPT_PROC: ; 1336
001C 00000 .WORD ^M<R2,R3,R4>
10BE SE 04 C2 00002  SUBL2 #4,SP
CF 00 FB 00005  CALLS #0,VERIFY_ISAM_DEFINITION ; 1362
03 50 E8 0000A  BLBS R0,+.3
0000V 31 0000D  BRW 39$
01 8F 9F 00010  PUSHAB #1 ; 1369
00000000G EF 01 FB 00013  CALLS #1,SCAN_DEFINITION
00000000G EF 00 D5 0001A  TSTL HIGH_KEY ; 1371
03 12 00020  BNEQ .+3
0000V 31 00022  BRW 37$
00000000 8F DF 00025  PUSHAL #0 ; 1378
78 8F 9F 0002B  PUSHAB #120
00000000G EF 9F 0002E  PUSHAB HIGH_KEY
0B 8F 9F 00034  PUSHAB #11
01 8F 9F 00037  PUSHAB #1
00000000G EF 05 FB 0003A  CALLS #5,FIND_OBJECT
00V 50 E9 00041  BLBC R0,4$
50 00000000G EF D0 00044  MOVL DEF_CURRENT,R0 ; 1380
```

	5C	27	A0	D0	0004B	MOVL	39(R0),LO_AREA	
			00V	11	0004F	BRB	5\$	
	5C		00	D2	00051	MCOML	#0,LO_AREA	: 1384
		00000000	8F	DF	00054	5\$: PUSHAL	#0	: 1386
		80	8F	9F	0005A	PUSHAB	#-128	
		00000000G	EF	9F	0005D	PUSHAB	HIGH_KEY	
		0B	8F	9F	00063	PUSHAB	#11	
		01	8F	9F	00066	PUSHAB	#1	
00000000G	EF		05	FB	00069	CALLS	#5,FIND_OBJECT	
	00V		50	E9	00070	BLBC	R0,7\$	
	50	00000000G	EF	D0	00073	MOVL	DEF_CURRENT,R0	: 1388
	52	27	A0	D0	0007A	MOVL	39(R0),L1_AREA	
			00V	11	0007E	BRB	8\$	
	52		00	D2	00080	7\$: MCOML	#0,L1_AREA	: 1392
		00000000	8F	DF	00083	8\$: PUSHAL	#0	: 1394
		7D	8F	9F	00089	PUSHAB	#125	
		00000000G	EF	9F	0008C	PUSHAB	HIGH_KEY	
		0B	8F	9F	00092	PUSHAB	#11	
		01	8F	9F	00095	PUSHAB	#1	
00000000G	EF		05	FB	00098	CALLS	#5,FIND_OBJECT	
	00V		50	E9	0009F	BLBC	R0,10\$	
	50	00000000G	EF	D0	000A2	MOVL	DEF_CURRENT,R0	: 1396
	53	27	A0	D0	000A9	MOVL	39(R0),LX_AREA	
			00V	11	000AD	BRB	11\$	
	53		00	D2	000AF	10\$: MCOML	#0,LX_AREA	: 1400
00000000G	EF	00000000G	EF	D0	000B2	11\$: MOVL	DEF_HEAD,DEF_CURRENT	: 1405
	54	00000000G	EF	D0	000BD	MOVL	DEF_CURRENT,R4	: 1407
	5C	27	A4	D1	000C4	13\$: CMPL	39(R4),LO_AREA	: 1413
			00V	12	000C8	BNEQ	18\$	
78	8F	1E	A4	91	000CA	CMPB	30(R4),#120	
			00V	12	000CF	BNEQ	18\$	
	0B	19	A4	91	000D1	CMPB	25(R4),#11	
			00V	12	000D5	BNEQ	18\$	
00000000G	EF	1A	A4	D1	000D7	CMPL	26(R4),HIGH_KEY	
			00V	13	000DF	BEQL	18\$	
	5C		00	D2	000E1	MCOML	#0,LO_AREA	: 1423
	52	27	A4	D1	000E4	18\$: CMPL	39(R4),L1_AREA	: 1425
			00V	12	000E8	BNEQ	23\$	
80	8F	1E	A4	91	000EA	CMPB	30(R4),#-128	
			00V	12	000EF	BNEQ	23\$	
	0B	19	A4	91	000F1	CMPB	25(R4),#11	
			00V	12	000F5	BNEQ	23\$	
00000000G	EF	1A	A4	D1	000F7	CMPL	26(R4),HIGH_KEY	
			00V	13	000FF	BEQL	23\$	
	52		00	D2	00101	MCOML	#0,L1_AREA	: 1435
	53	27	A4	D1	00104	23\$: CMPL	39(R4),LX_AREA	: 1437
			00V	12	00108	BNEQ	28\$	
7D	8F	1E	A4	91	0010A	CMPB	30(R4),#125	
			00V	12	0010F	BNEQ	28\$	
	0B	19	A4	91	00111	CMPB	25(R4),#11	
			00V	12	00115	BNEQ	28\$	
00000000G	EF	1A	A4	D1	00117	CMPL	26(R4),HIGH_KEY	
			00V	13	0011F	BEQL	28\$	
	53		00	D2	00121	MCOML	#0,LX_AREA	: 1447
00000000G	EF		00	FB	00124	28\$: CALLS	#0,INCR_CURRENT	: 1449
		00000000G	EF	D5	0012B	TSTL	DEF_CURRENT	
			91	12	00131	BNEQ	13\$	

Generated Code								
		00000000G	EF	9F	00133	PUSHAB	HIGH_KEY	: 1458
		0B	8F	9F	00139	PUSHAB	#11	
0000V	CF		02	FB	0013C	CALLS	#2,DELETE_SECTION	
			5C	D5	00141	TSTL	L0_AREA	: 1463
			00V	19	00143	BLSS	31\$	
FC	AD		5C	D0	00145	MOVL	L0_AREA,-4(FP)	: 1465
		FC	AD	9F	00149	PUSHAB	-4(FP)	
		05	8F	9F	0014C	PUSHAB	#5	
0000V	CF		02	FB	0014F	CALLS	#2,DELETE_SECTION	
			52	D5	00154	TSTL	L1_AREA	: 1467
			00V	19	00156	BLSS	33\$	
FC	AD		52	D0	00158	MOVL	L1_AREA,-4(FP)	: 1469
		FC	AD	9F	0015C	PUSHAB	-4(FP)	
		05	8F	9F	0015F	PUSHAB	#5	
0000V	CF		02	FB	00162	CALLS	#2,DELETE_SECTION	
			53	D5	00167	TSTL	LX_AREA	: 1471
			00V	19	00169	BLSS	35\$	
FC	AD		53	D0	0016B	MOVL	LX_AREA,-4(FP)	: 1473
		FC	AD	9F	0016F	PUSHAB	-4(FP)	
		05	8F	9F	00172	PUSHAB	#5	
0000V	CF		02	FB	00175	CALLS	#2,DELETE_SECTION	
		00000000G	EF	9F	0017A	PUSHAB	SHIFT	: 1475
			04	DD	00180	PUSHL	#4	
		00000000G	EF	9F	00182	PUSHAB	PASSFV_OUTPUT	
00000000G	EF	FFFFED95	03	FB	00188	CALLS	#3,PASSWRITE_STRING	
			EF	9F	0018F	PUSHAB	C.AAK	
			21	DD	00195	PUSHL	#33	
		00000000G	EF	9F	00197	PUSHAB	PASSFV_OUTPUT	
00000000G	EF		03	FB	0019D	CALLS	#3,PASSWRITE_STRING	
		00000000G	EF	9F	001A4	PUSHAB	PASSFV_OUTPUT	
00000000G	EF		01	FB	001AA	CALLS	#1,PASSWriteln2	
		0000001F	8F	DF	001B1	PUSHAL	#31	: 1476
00000000G	EF		01	FB	001B7	CALLS	#1,QUERY	
			00V	11	001BE	BRB	38\$	
		00000000G	EF	9F	001C0	PUSHAB	SHIFT	: 1484
			04	DD	001C6	PUSHL	#4	
		00000000G	EF	9F	001C8	PUSHAB	PASSFV_OUTPUT	
00000000G	EF		03	FB	001CE	CALLS	#3,PASSWRITE_STRING	
		00000000G	EF	9F	001D5	PUSHAB	ANSI_REVERSE	
			04	DD	001DB	PUSHL	#4	
		00000000G	EF	9F	001DD	PUSHAB	PASSFV_OUTPUT	
00000000G	EF		03	FB	001E3	CALLS	#3,PASSWRITE_STRING	
		FFFFED5E	EF	9F	001EA	PUSHAB	C.AAL	
			2E	DD	001F0	PUSHL	#46	
		00000000G	EF	9F	001F2	PUSHAB	PASSFV_OUTPUT	
00000000G	EF		03	FB	001F8	CALLS	#3,PASSWRITE_STRING	
		00000000G	EF	9F	001FF	PUSHAB	ANSI_RESET	
			04	DD	00205	PUSHL	#4	
		00000000G	EF	9F	00207	PUSHAB	PASSFV_OUTPUT	
00000000G	EF		03	FB	0020D	CALLS	#3,PASSWRITE_STRING	
		00000000G	EF	9F	00214	PUSHAB	PASSFV_OUTPUT	
00000000G	EF		01	FB	0021A	CALLS	#1,PASSWriteln2	
		00004140	8F	DF	00221	PUSHAF	#^F3.0	: 1488
00000000G	EF		01	FB	00227	CALLS	#1,LIB\$WAIT	
				0022E	38\$:			
			04	0022E	39\$:	RET		: 1494

; Routine Size: 559 bytes, Routine Base: \$CODE + 01272

			00000	DELETE_SECTION:		; 1338
			0004 00000	.WORD	^M<R2>	
	5E		08 C2 00002	SUBL2	#8,SP	
	52	04	BC 90 00005	MOVB	@4(R12),SECTION	
	5C	08	BC D0 00009	MOVL	@8(R12),SECT_NUM	
		00000000	8F DF 0000D	PUSHAL	#0	; 1342
		00	8F 9F 00013	PUSHAB	#0	
FC	AD		5C D0 00016	MOVL	SECT_NUM,-4(FP)	
		FC	AD 9F 0001A	PUSHAB	-4(FP)	
F8	AD		52 90 0001D	MOVB	SECTION,-8(FP)	
		F8	AD 9F 00021	PUSHAB	-8(FP)	
		00	8F 9F 00024	PUSHAB	#0	
00000000G	EF		05 FB 00027	CALLS	#5,FIND_OBJECT	
	03		50 E8 0002E	BLBS	R0,..+3	
		0000V	31 00031	BRW	3\$	
		00000000G	EF 9F 00034	PUSHAB	SHIFT	; 1346
			04 DD 0003A	PUSHL	#4	
		00000000G	EF 9F 0003C	PUSHAB	PASSFV OUTPUT	
00000000G	EF		03 FB 00042	CALLS	#3,PASSWRITE_STRING	
		FFFFED00	EF 9F 00049	PUSHAB	C.AAM	
			09 DD 0004F	PUSHL	#9	
		00000000G	EF 9F 00051	PUSHAB	PASSFV OUTPUT	
00000000G	EF		03 FB 00057	CALLS	#3,PASSWRITE_STRING	
	50		52 9A 0005E	MOVZBL	SECTION,R0	
7E		00000000GEF	40 9A 00061	MOVZBL	PRIMARY_WIDTH[R0],-(SP)	
7E			52 9A 00069	MOVZBL	SECTION,-(SP)	
		FFFCE9	EF 9F 0006C	PUSHAB	C.AAN	
		00000000G	EF 9F 00072	PUSHAB	PASSFV OUTPUT	
00000000G	EF		04 FB 00078	CALLS	#4,PASSWRITE_ENUMERATED	
			03 DD 0007F	PUSHL	#3	
			5C DD 00081	PUSHL	SECT_NUM	
		00000000G	EF 9F 00083	PUSHAB	PASSFV OUTPUT	
00000000G	EF		03 FB 00089	CALLS	#3,PASSWRITE_INTEGER	
		FFFFED9D	EF 9F 00090	PUSHAB	C.AAO	
			11 DD 00096	PUSHL	#17	
		00000000G	EF 9F 00098	PUSHAB	PASSFV OUTPUT	
00000000G	EF		03 FB 0009E	CALLS	#3,PASSWRITE_STRING	
		00000000G	EF 9F 000A5	PUSHAB	PASSFV OUTPUT	
00000000G	EF		01 FB 000AB	CALLS	#1,PASSWRITELN2	
		0000001F	8F DF 000B2	PUSHAL	#31	; 1348
00000000G	EF		01 FB 000B8	CALLS	#1,QUERY	
	FC		5C D0 000BF	MOVL	SECT_NUM,-4(FP)	; 1349
		FC	AD 9F 000C3	PUSHAB	-4(FP)	
	F8		52 90 000C6	MOVB	SECTION,-8(FP)	
		F8	AD 9F 000CA	PUSHAB	-8(FP)	
00000000G	EF		02 FB 000CD	CALLS	#2,DELETE_PRIMARY_SECTION	; 1353
			04 000D4	RET		

; Routine Size: 213 bytes, Routine Base: \$CODE + 014A1

			00000	OPTIMIZE_SCRIPT_PROC:		; 1540
			0000 00000	.WORD	^M<>	
03	00000000G	EF	00 E1 00002	BBC	#0,ANALYSIS_SPECIFIED,..+3	; 1547
		0000V	31 0000A	BRW	6\$	
00V00000000G	EF		00 E0 0000D	BBS	#0,AUTO_TUNE,3\$; 1550

Generated Code							
		00000000G	EF	9F	00015	PUSHAB	SHIFT ; 1551
			04	DD	0001B	PUSHL	#4
		00000000G	EF	9F	0001D	PUSHAB	PASSFV OUTPUT
00000000G	EF	FFFFED42	03	FB	00023	CALLS	#3,PASSWRITE_STRING
			EF	9F	0002A	PUSHAB	C.AAP
			38	DD	00030	PUSHL	#56
		00000000G	EF	9F	00032	PUSHAB	PASSFV OUTPUT
00000000G	EF		03	FB	00038	CALLS	#3,PASSWRITE_STRING
		00000000G	EF	9F	0003F	PUSHAB	CRLF_SHIFT
			06	DD	00045	PUSHL	#6
		00000000G	EF	9F	00047	PUSHAB	PASSFV OUTPUT
00000000G	EF		03	FB	0004D	CALLS	#3,PASSWRITE_STRING
		00000000G	EF	9F	00054	PUSHAB	PASSFV OUTPUT
00000000G	EF		01	FB	0005A	CALLS	#1,PASSWriteln2
			00V	11	00061	BRB	4\$
			00	DD	00063	PUSHL	#0 ; 1557
			00	DD	00065	PUSHL	#0
			00	DD	00067	PUSHL	#0
		00B3801C	8F	DD	00069	PUSHL	#11763740
00000000G	EF		04	FB	0006F	CALLS	#4,LIB\$STOP
00000000G	EF		01	90	00076	MOVB	#1,VISIBLE_QUESTION ; 1559
		00000000E	8F	DF	0007D	PUSHAL	#14 ; 1561
00000000G	EF		01	FB	00083	CALLS	#1,QUERY
		00000000G	EF	94	0008A	CLRB	VISIBLE_QUESTION ; 1563
00000000G	EF		01	90	00090	MOVB	#1,ANALYSIS_SPECIFIED ; 1565
00000000G	EF		00	FB	00097	CALLS	#0,INPUT_ANALYSIS_FILE ; 1569
			5C	94	0009E	CLRB	AN_KEY_FOUND ; 1571
00000000G	EF		00	FB	000A0	CALLS	#0,POINT_AT_ANALYSIS ; 1573
00000000G	EF	00000000G	EF	D0	000A7	MOVL	DEF_HEAD,DEF_CURRENT ; 1575
	50	00000000G	EF	D0	000B2	MOVL	DEF_CURRENT,R0 ; 1579
	04	19	A0	91	000B9	CMPB	25(R0),#4
			00V	12	000BD	BNEQ	9\$
	5C		01	90	000BF	MOVB	#1,AN_KEY_FOUND ; 1581
00000000G	EF		00	FB	000C2	CALLS	#0,INCR_CURRENT ; 1583
	00V		5C	E8	000C9	BLBS	AN_KEY_FOUND,11\$
		00000000G	EF	D5	000CC	TSTL	DEF_CURRENT
			DE	12	000D2	BNEQ	7\$
00000000G	EF		00	FB	000D4	CALLS	#0,POINT_AT_DEFINITION ; 1587
	00V		5C	E9	000DB	BLBC	AN_KEY_FOUND,13\$; 1589
00000000G	EF		01	90	000DE	MOVB	#1,OPTIMIZING ; 1593
121F	CF		00	FB	000E5	CALLS	#0,REDESIGN_SCRIPT_PROC ; 1594
		0000V	31	000EA	BRW	16\$	
00V00000000G	EF		00	E0	000ED	BBS	#0,AUTO_TUNE,16\$; 1602
		00000000G	EF	9F	000F5	PUSHAB	SHIFT ; 1604
			04	DD	000FB	PUSHL	#4
		00000000G	EF	9F	000FD	PUSHAB	PASSFV OUTPUT
00000000G	EF		03	FB	00103	CALLS	#3,PASSWRITE_STRING
		FFFFEC9A	EF	9F	0010A	PUSHAB	C.AAQ
		00000040	8F	DD	00110	PUSHL	#64
		00000000G	EF	9F	00116	PUSHAB	PASSFV OUTPUT
00000000G	EF		03	FB	0011C	CALLS	#3,PASSWRITE_STRING
		00000000G	EF	9F	00123	PUSHAB	PASSFV OUTPUT
00000000G	EF		01	FB	00129	CALLS	#1,PASSWriteln2
		00000000G	EF	9F	00130	PUSHAB	SHIFT
			04	DD	00136	PUSHL	#4 ; 1606
		00000000G	EF	9F	00138	PUSHAB	PASSFV OUTPUT
00000000G	EF		03	FB	0013E	CALLS	#3,PASSWRITE_STRING

		FFFFEC9F	EF	9F	00145	PUSHAB	C.AAR		
			3F	DD	0014B	PUSHL	#63		
00000000G	EF	00000000G	EF	9F	0014D	PUSHAB	PASSFV OUTPUT		
		00000000G	03	FB	00153	CALLS	#3,PASSWRITE_STRING		
00000000G	EF	00000000G	EF	9F	0015A	PUSHAB	PASSFV OUTPUT		
		00000002	01	FB	00160	CALLS	#1,PASSWRITELN2		
00000000G	EF		8F	DF	00167	PUSHAL	#2		: 1609
		00000000G	01	FB	0016D	CALLS	#1,CLEAR		
			EF	94	00174	CLRB	OPTIMIZING		: 1613
				04	0017A	RET			: 1615

; Routine Size: 379 bytes, Routine Base: \$CODE + 01576

				00000	INVOKE_SCRIPT:				: 1664
				0000	00000	.WORD	^M<>		
00000008G	EF		07	DD	00002	MOVL	#7, IDATA+8		: 1671
	07	00000108G	EF	D1	00009	CMPL	IDATA+264, #7		: 1676
			00V	12	00010	BNEQ	3\$		
		00000042	8F	DF	00012	PUSHAL	#66		: 1683
00000000G	EF		01	FB	00018	CALLS	#1, QUERY		
			0000V	31	0001F	BRW	15\$		
03 00000000G	EF		00	E1	00022	BBC	#0, AUTO_TUNE, .+3		: 1691
			0000V	31	0002A	BRW	15\$		
		00000003	8F	DF	0002D	PUSHAL	#3		: 1695
00000000G	EF		01	FB	00033	CALLS	#1, CLEAR		
		00000000G	EF	9F	0003A	PUSHAB	SHIFT		: 1696
			04	DD	00040	PUSHL	#4		
		00000000G	EF	9F	00042	PUSHAB	PASSFV OUTPUT		
00000000G	EF		03	FB	00048	CALLS	#3, PASSWRITE_STRING		
			01	DD	0004F	PUSHL	#1		
	7E	00000000G	EF	9A	00051	MOVZBL	TAB, -(SP)		
		00000000G	EF	9F	00058	PUSHAB	PASSFV OUTPUT		
00000000G	EF		03	FB	0005E	CALLS	#3, PASSWRITE_CHAR		
			01	DD	00065	PUSHL	#1		
	7E	00000000G	EF	9A	00067	MOVZBL	TAB, -(SP)		
		00000000G	EF	9F	0006E	PUSHAB	PASSFV OUTPUT		
00000000G	EF		03	FB	00074	CALLS	#3, PASSWRITE_CHAR		
		00000000G	EF	9F	0007B	PUSHAB	ANSI_REVERSE		
			04	DD	00081	PUSHL	#4		
		00000000G	EF	9F	00083	PUSHAB	PASSFV OUTPUT		
06 00000000G	EF		03	FB	00089	CALLS	#3, PASSWRITE_STRING		
	00	00000108G	EF	CF	00090	CASEL	IDATA+264, #0, #6		: 1698
			0000V		00098	.DISPL	5\$		
			0000V		0009A	.DISPL	6\$		
			0000V		0009C	.DISPL	7\$		
			0000V		0009E	.DISPL	9\$		
			0000V		000A0	.DISPL	8\$		
			0000V		000A2	.DISPL	10\$		
			0000V		000A4	.DISPL	11\$		
			0000V	31	000A6	BRW	12\$		
		FFFFEC00	EF	9F	000A9	PUSHAB	C.AAS		: 1700
			08	DD	000AF	PUSHL	#8		
		00000000G	EF	9F	000B1	PUSHAB	PASSFV OUTPUT		
00000000G	EF		03	FB	000B7	CALLS	#3, PASSWRITE_STRING		
			0000V	31	000BE	BRW	13\$		
		FFFFEBF0	EF	9F	000C1	PUSHAB	C.AAT		: 1701
			0B	DD	000C7	PUSHL	#11		

Generated Code							
00000000G	EF	00000000G	EF	9F	000C9	PUSHAB	PASSFV OUTPUT
			03	FB	000CF	CALLS	#3,PASSWRITE_STRING
			00V	11	000D6	BRB	13\$
		FFFFEBE5	EF	9F	000D8	7\$: PUSHAB	C.AAU ; 1702
			08	DD	000DE	PUSHL	#8
00000000G	EF	00000000G	EF	9F	000E0	PUSHAB	PASSFV OUTPUT
			03	FB	000E6	CALLS	#3,PASSWRITE_STRING
			00V	11	000ED	BRB	13\$
		FFFFEBD6	EF	9F	000EF	8\$: PUSHAB	C.AAV ; 1703
			0B	DD	000F5	PUSHL	#11
00000000G	EF	00000000G	EF	9F	000F7	PUSHAB	PASSFV OUTPUT
			03	FB	000FD	CALLS	#3,PASSWRITE_STRING
			00V	11	00104	BRB	13\$
		FFFFEBCB	EF	9F	00106	9\$: PUSHAB	C.AAW ; 1704
			09	DD	0010C	PUSHL	#9
00000000G	EF	00000000G	EF	9F	0010E	PUSHAB	PASSFV OUTPUT
			03	FB	00114	CALLS	#3,PASSWRITE_STRING
			00V	11	0011B	BRB	13\$
		FFFFEBC0	EF	9F	0011D	10\$: PUSHAB	C.AAX ; 1705
			09	DD	00123	PUSHL	#9
00000000G	EF	00000000G	EF	9F	00125	PUSHAB	PASSFV OUTPUT
			03	FB	0012B	CALLS	#3,PASSWRITE_STRING
			00V	11	00132	BRB	13\$
		FFFFEBB5	EF	9F	00134	11\$: PUSHAB	C.AAY ; 1706
			08	DD	0013A	PUSHL	#8
00000000G	EF	00000000G	EF	9F	0013C	PUSHAB	PASSFV OUTPUT
			03	FB	00142	CALLS	#3,PASSWRITE_STRING
			00V	11	00149	BRB	13\$
					0014B	12\$:	
		FFFFEBA6	EF	9F	0014B	13\$: PUSHAB	C.AAZ ; 1714
			08	DD	00151	PUSHL	#8
00000000G	EF	00000000G	EF	9F	00153	PUSHAB	PASSFV OUTPUT
			03	FB	00159	CALLS	#3,PASSWRITE_STRING
		00000000G	EF	9F	00160	PUSHAB	ANSI_RESET
			04	DD	00166	PUSHL	#4
00000000G	EF	00000000G	EF	9F	00168	PUSHAB	PASSFV OUTPUT
			03	FB	0016E	CALLS	#3,PASSWRITE_STRING
		00000000G	EF	9F	00175	PUSHAB	CRLF
			02	DD	0017B	PUSHL	#2
00000000G	EF	00000000G	EF	9F	0017D	PUSHAB	PASSFV OUTPUT
			03	FB	00183	CALLS	#3,PASSWRITE_STRING
		00000000G	EF	9F	0018A	PUSHAB	PASSFV OUTPUT
00000000G	EF		01	FB	00190	CALLS	#1,PASSWriteln2
00000000G	EF		01	90	00197	15\$: MOVB	#1,TAKE_DEFAULTS ; 1720
06	00	00000108G	EF	CF	0019E	CASEL	IDATA+264,#0,#6 ; 1722
			0000V		001A6	.DISPL	19\$
			0000V		001A8	.DISPL	20\$
			0000V		001AA	.DISPL	16\$
			0000V		001AC	.DISPL	18\$
			0000V		001AE	.DISPL	17\$
			0000V		001B0	.DISPL	21\$
			0000V		001B2	.DISPL	22\$
			0000V		001B4	BRW	23\$
00000000G	EF		00	FB	001B7	16\$: CALLS	#0,WARN_OF_ERASE ; 1728
00000000G	EF		00	FB	001BE	CALLS	#0,INIT_DEF ; 1729
		00	8F	9F	001C5	PUSHAB	#0 ; 1730
		00	8F	9F	001C8	PUSHAB	#0

Generated Code

00000000G	EF	02	FB	001CB	CALLS	#2, INDEXED_DESIGN	
		00V	11	001D2	BRB	24\$	
00000000G	EF	00	FB	001D4	17\$: CALLS	#0, WARN_OF_ERASE	: 1738
00000000G	EF	00	FB	001DB	CALLS	#0, INIT_DEF	: 1739
00000000G	EF	00	FB	001E2	CALLS	#0, SEQ_REL_WORK	: 1740
00000000G	EF	00	FB	001E9	CALLS	#0, SEQ_DEF	: 1741
		00V	11	001F0	BRB	24\$	
00000000G	EF	00	FB	001F2	18\$: CALLS	#0, WARN_OF_ERASE	: 1749
00000000G	EF	00	FB	001F9	CALLS	#0, INIT_DEF	: 1750
00000000G	EF	00	FB	00200	CALLS	#0, SEQ_REL_WORK	: 1751
00000000G	EF	00	FB	00207	CALLS	#0, REL_DEF	: 1752
		00V	11	0020E	BRB	24\$	
1237	CF	00	FB	00210	19\$: CALLS	#0, ADD_KEY_SCRIPT_PROC	: 1756
		00V	11	00215	BRB	24\$	
1272	CF	00	FB	00217	20\$: CALLS	#0, DELETE_KEY_SCRIPT_PROC	: 1758
		00V	11	0021C	BRB	24\$	
1576	CF	00	FB	0021E	21\$: CALLS	#0, OPTIMIZE_SCRIPT_PROC	: 1760
		00V	11	00223	BRB	24\$	
121F	CF	00	FB	00225	22\$: CALLS	#0, REDESIGN_SCRIPT_PROC	: 1762
		00V	11	0022A	BRB	24\$	
				0022C	23\$: CLRB	TAKE_DEFAULTS	: 1770
00000000G	EF	94	0022C	24\$: RET			: 1772
		04	00232				

; Routine Size: 563 bytes, Routine Base: \$CODE + 016F1

				00000	SET_PROC:		: 1817
				0000	00000		
				01	90	00002	
00000000G	EF	00000043	8F	DF	00009	MOVW	#1, VISIBLE_QUESTION
						PUSHAL	#67
00000000G	EF	0000010CG	01	FB	0000F	CALLS	#1, QUERY
07	00		EF	CF	00016	CASEL	IDATA+268, #0, #7
			0000V		0001E	.DISPL	12\$
			0000V		00020	.DISPL	2\$
			0000V		00022	.DISPL	4\$
			0000V		00024	.DISPL	6\$
			0000V		00026	.DISPL	16\$
			0000V		00028	.DISPL	14\$
			0000V		0002A	.DISPL	10\$
			0000V		0002C	.DISPL	8\$
			0000V	31	0002E	BRW	18\$
00000000G	EF	00000046	8F	DF	00031	2\$: PUSHAL	#70
			01	FB	00037	CALLS	#1, QUERY
			00V	11	0003E	BRB	19\$
00000000G	EF	00000026	8F	DF	00040	4\$: PUSHAL	#38
			01	FB	00046	CALLS	#1, QUERY
			00V	11	0004D	BRB	19\$
00000000G	EF	0000002F	8F	DF	0004F	6\$: PUSHAL	#47
			01	FB	00055	CALLS	#1, QUERY
			00V	11	0005C	BRB	19\$
00000000G	EF	00000041	8F	DF	0005E	8\$: PUSHAL	#65
			01	FB	00064	CALLS	#1, QUERY
			00V	11	0006B	BRB	19\$
00000000G	EF	0000003F	8F	DF	0006D	10\$: PUSHAL	#63
			01	FB	00073	CALLS	#1, QUERY
			00V	11	0007A	BRB	19\$
			8F	DF	0007C	12\$: PUSHAL	#14

EDFFUNCS
V04-000

J 3
16-Sep-1984 01:17:14
5-Sep-1984 13:37:08

VAX-11 Pascal V2.4-277
DISK\$VMSMASTER:[EDF.SRC]EDFFUNCS.PAS;1 (28) Page 65

Generated Code							
00000000G	EF	01	FB	00082	CALLS	#1, QUERY	
		00V	11	00089	BRB	19\$	
	0000000F	8F	DF	0008B	PUSHAL	#15	; 1833
00000000G	EF	01	FB	00091	CALLS	#1, QUERY	
		00V	11	00098	BRB	19\$	
	0000003C	8F	DF	0009A	PUSHAL	#60	; 1839
00000000G	EF	01	FB	000A0	CALLS	#1, QUERY	
00000000G	EF	01	90	000A7	MOVB	#1, NUMBER_KEYS_SET	; 1840
		00V	11	000AE	BRB	19\$	
				000B0			
	00000000G	EF	94	000B0	CLRB	VISIBLE_QUESTION	; 1850
			04	000B6	RET		; 1852

; Routine Size: 183 bytes, Routine Base: \$CODE + 01924

019DB .END

EDFFUNCS
V04-000

Pascal Compilation Statistics

K 3
16-Sep-1984 01:17:14
5-Sep-1984 13:37:08

VAX-11 Pascal V2.4-277
DISK\$VMSMASTER:[EDF.SRC]EDFFUNCS.PAS;1 (28) Page 66

COMMAND QUALIFIERS

PASCAL/MACHINE/NODEBUG/NOCHECK/LIS=LIS\$:EDFFUNCS/OBJ=OBJ\$:EDFFUNCS MSRC\$:EDFFUNCS

/CHECK=(NOBOUNDS, NOCASE_SELECTORS, NOOVERFLOW, NOPOINTERS, NOSUBRANGE)

/DEBUG=(NOSYMBOLS, NOTRACEBACK)

/ENVIRONMENT= \$255\$DUA28:[EDF.OBJ]EDFFUNCS.PEN;1

/LIST= \$255\$DUA28:[EDF.LIS]EDFFUNCS.LIS;1

/OBJECT= \$255\$DUA28:[EDF.OBJ]EDFFUNCS.OBJ;1

/NOCROSS_REFERENCE /ERROR_LIMIT=30 /NOG_FLOATING /MACHINE_CODE /NOOLD_VERSION /OPTIMIZE /NOSTANDARD /WARNINGS

COMPILER INTERNAL TIMING

Phase	Faults	CPU Time	Elapsed Time
Initialization	85	00:00.4	00:02.7
Source Analysis	1093	00:19.1	04:40.7
Source Listing	81	00:02.3	00:07.1
Tree Construction	236	00:01.1	00:02.6
Flow Analysis	24	00:00.5	00:01.0
Profit Analysis	41	00:00.7	00:02.2
Context Analysis	222	00:06.5	00:12.4
Name Packing	2	00:00.3	00:00.7
Code Selection	19	00:01.4	00:03.5
Final	217	00:05.8	00:16.1
TOTAL	2026	00:38.1	05:29.0

COMPILATION STATISTICS

CPU Time: 00:38.1 (2921 Lines/Minute)
Elapsed Time: 05:29.0
Page Faults: 2026
Compilation Complete

0126

AH-BT13A-SE
 VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

0127 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

